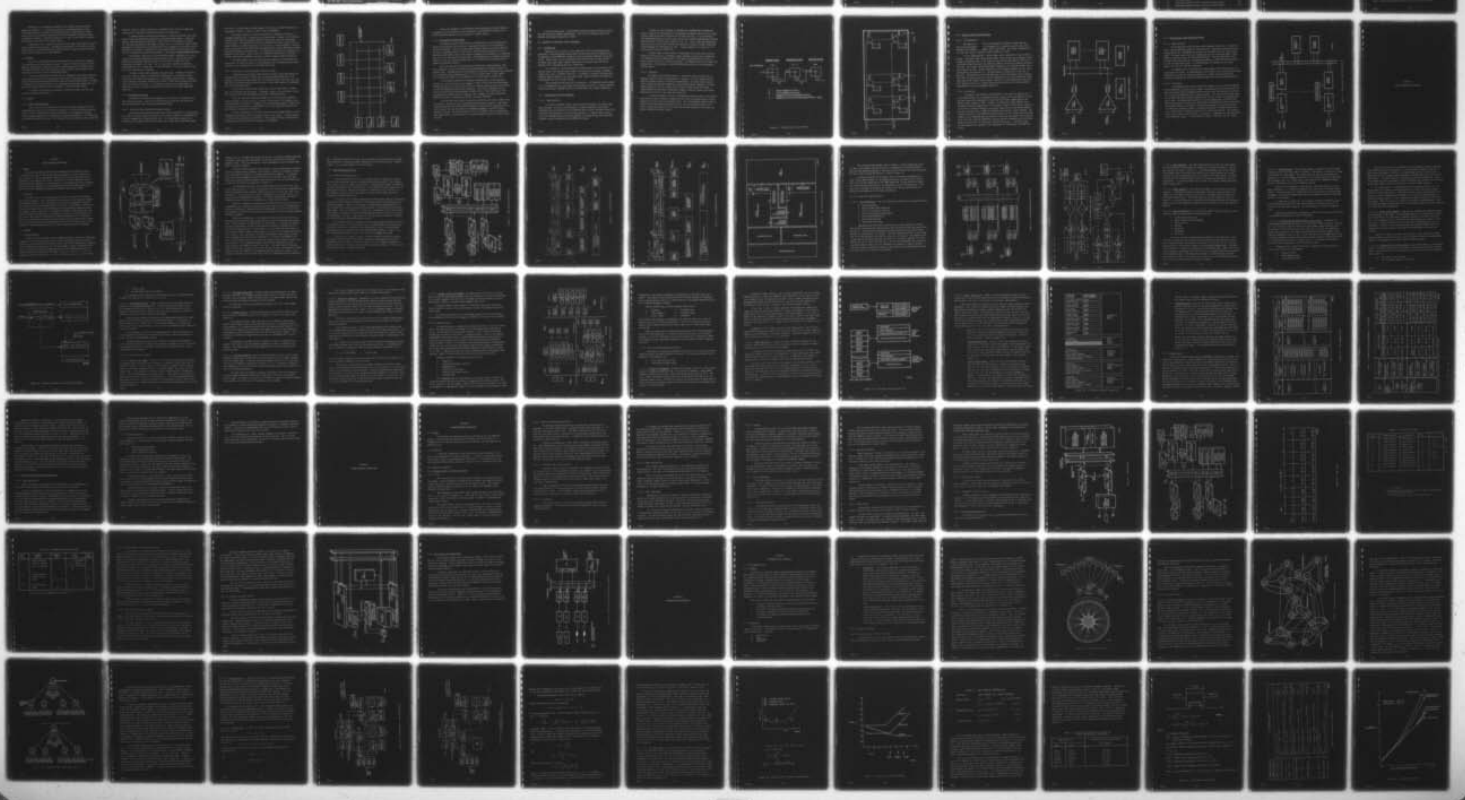
AD-A039 549 GTE SYLVANIA INC NEEDHAM HEIGHTS MASS ELECTRONIC SYS--ETC F/G 17/2
SENET-DAX STUDY. VOLUME 2.(U)
JUN 76 DCA100-75-C-0071
UNCLASSIFIED FR76-1-VOL-2 NL

1 OF 3
AD A039549



FINAL REPORT

SENET-DAX STUDY VOL. 2

25 JUNE 1976

GTE SYLVANIA INCORPORATED
ELECTRONIC SYSTEMS GROUP
EASTERN DIVISION

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): ~~FINAL REPORT~~ SENET-DAX STUDY. Volume 2. ~~VOL. 1 AND VOL. 2~~	5. TYPE OF REPORT & PERIOD COVERED: Final Report. June 75 — June 76.
	6. PERFORMING ORG. REPORT NUMBER: FR76-1-Vol-2
7. AUTHOR(s)	8. CONTRACT OR GRANT NUMBER(s): DCA100-75-C-0071
9. PERFORMING ORGANIZATION NAME AND ADDRESS: GTE Sylvania Inc. Electronic Systems Group, Eastern Division Needham Heights, Massachusetts 02194	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: PE 33143K Task 13103G
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Systems Concepts Branch, R740 Defense Communications Engineering Center Reston, Virginia 22090	12. REPORT DATE: 25 June 1976
	13. NUMBER OF PAGES: 490 (12) 262 p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same as 11	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Distribution unlimited.

18. SUPPLEMENTARY NOTES

None.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Integrated voice/data switching
Time division multiplexing
Dynamic channel allocation
Digital Access Exchange (DAX)
Distributed Processor Architecture

406451

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The SENET-DAX study investigated the concept of an integrated voice/data switching system using time division multiplexing and dynamic channel allocation; the objectives being: (1) To analyze the feasibility of the SENET-DAX concept in terms of structure and performance characteristic. (2) To address the translation of the concept into candidate hardware and software techniques that could be used to implement the concept. The results of the analysis and techniques studies is presented in these two volumes.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SENET-DAX STUDY
FINAL REPORT

Volume 2

(Contract No. DCA-100-75-C-0071)

25 June 1976

Submitted to

Defense Communications Engineering Center
Defense Communications Agency
Washington, D.C.

GTE SYLVANIA
INCORPORATED
ELECTRONIC SYSTEMS GROUP
EASTERN DIVISION

77 "A" STREET
NEEDHAM HEIGHTS, MASSACHUSETTS 02194

VOLUME 2

TABLE OF CONTENTS

→ to iv

from iii

VOLUME 2

TABLE OF CONTENTS (Cont)

VOLUME 2

TABLE OF CONTENTS (Cont)

VOLUME 2

LIST OF ILLUSTRATIONS

VOLUME 2

LIST OF ILLUSTRATIONS (Cont)

LIST OF ILLUSTRATIONS (Cont)

VOLUME 2

LIST OF TABLES

SECTION 6

SELECTION OF SYSTEM PROCESSOR ARCHITECTURE

SECTION 6

SELECTION OF SYSTEM PROCESSOR ARCHITECTURE

6.1 PROBLEM

The purpose of this section is to discuss the trade-off analysis of different approaches to implementation of the processing system architecture for the SENET-DAX System. The various approaches are to be evaluated against the requirements imposed by the SENET-DAX communications concept. The basic concept, full bandwidth utilization by time domain allocations of transmitted digitized information, is applied to an integrated voice/data multi-node switching network wherein multiplexing and switching functions are accomplished by internal computer processes.

6.2 OBJECTIVES

In considering SENET-DAX architecture, it has been our intent to maximize the use of stored-program control in order to meet the extraordinary flexibility goals required by the dynamic allocation approach. In addition, the architecture recommended should not only implement the flexibility of the SENET-DAX concept, but allow its extension to processing of higher transmission rates that will result from evolutionary developments in solid-state technology, particularly in higher switching speeds for memory and logic.

Further objectives in the tradeoff analysis were in the areas of diagnostic capability, the use of developing technology, and system reliability and flexibility. The recommended architecture should provide a significantly higher level of diagnostic capabilities than are currently implemented in communication switching systems. The architecture must utilize, consistent with other goals, integrated circuit technology to simplify hardware and software design and to reduce the complexity of the overall software operating system. Lastly, the objectives of reliability, modularity and expandability as applied to military networks must be met. The past point merits more discussion here.

Reliability is of paramount concern, since this aspect contributes greatly to system cost and the system should be structured to minimize redundancy costs without impairment of reliability. Component multiplicities will occur by system nature, and maximum advantage should be taken of all such instances.

Modularity is an essential ingredient of any complex system with high diagnostic resolution. It contributes greatly to the maintainability of the system and therefore is related to system availability. Ease of fault location and the ability to gracefully degrade service have a profound simplifying effect on diagnostic software. Diagnostic software is as costly as any other to develop, and modularity will enhance its simplification.

Expandibility is an important attribute of any large system that will experience an unknown growth pattern. Configuration flexibility, as a function of site requirements, will provide additional control of total system costs while allowing natural growth without major system upheaval.

6.3 APPROACH

Only those system processing architectures that are relevant to communication switching systems have been evaluated. A number of other architectures that optimize computational capabilities or off-line data processing, or both, are not considered relevant, since the DAX concept caters primarily to real-time traffic requiring very little (if any) computational power.

Processor architectures examined from the standpoint of the above objectives were single processors, multi-processors with physical or functional partitioning, and distributed processors with both functional and physical partitioning. It should be emphasized that the architectures are evolutionary rather than competitive, and that each shows strengths as well as weaknesses in any given application.

In addition to processor architectures, alternative matrix architectures are also considered (Section 6.5).

6.4 PROGRESS

6.4.1 Single Processor System

This is the classic centralized "all-in-one processor" approach, which is characterized by one set of instructions operating on one set of data at any given time. In this centralized processing architecture, the throughput or maximum traffic handling capability is fundamentally governed and limited by the instruction execution and memory cycle times. For considerations of reliability, the centralized

processor system is usually provided on a redundant basis, with the standby unit taking over all functions in case of failure of the primary unit.

A single processor architecture represents a significant advance over wired logic systems in providing the flexibility of stored program control. There are, however, some important considerations. While there is an advantage to having the same processor handle a wide range of switch sizes, the relatively high overhead processing cost often makes the application of this technique for the smaller sizes prohibitively expensive. Secondly, the centralized monolithic architecture often results in complicated design and maintenance of the overall software operating system.

With specific application to the SENET-DAX concept, examination of a large central or monolithic processing structure revealed that no currently-available medium-to-large general purpose processor has the speed to handle the total processing requirements of a node. This is especially true when considering a tandem node with 14 T1 trunks and up to 600 subscribers, with some terminals (slow-scan-video) operating at 200 Kb/sec.

A further comment about redundancy should be made. Redundant centralized processors, with or without a load-sharing configuration, may introduce higher system costs and additional complexities in the program design and will certainly require automatic switchover circuitry and control. While these problems have been solved many times in the past, it is necessary, because of the requirements of SENET-DAX processing, to look at processor architectures that reduce operational complexity.

6.4.2 Multiprocessor Systems

The advent of economical mini-computers led to practical consideration of multi-processor systems. These can be divided into systems that are physically partitioned and those that are functionally partitioned.

6.4.2.1 Physically Partitioned Multi-Processor System

A physically partitioned multi-processor system architecture is characterized by a number of general purpose processors, each processor capable of performing any function commanded by the Common Control Unit. All individual processors are scheduled by the Common Control Unit and they share common memory and input-output

facilities. Systems capable of mode changes in this way are often referred to as "polymorphic" systems. Figure 6-1 shows such an arrangement.

The primary impetus to such an architecture is economical increase in throughput compared with a single processor architecture. However, it is difficult to realize this complex switching function without undue costs for centralized switch control. Introduction of centralized switching and control tends to have a negative effect on system reliability, not only due to its own complexity, but also to re-introduction of the single thread switching element, which immediately imposes additional redundancy considerations. True polymorphic systems also complicate the operating system design, and can result in elaborate sophistication in fallback, failure, and recovery procedures in order to achieve higher system availability.

The higher complexity of this architecture compared with distributed processing does not appear to result in any advantage over a distributed architecture (Section 6.4.3) for application to the SENET-DAX system.

6.4.2.2 Functionally Partitioned Multi-processor System

A functionally partitioned multi-processor system is generally implied to mean processor(s) assigned to handle only certain pre-defined functions. For instance, one processor interfacing with the central processor unit (CPU) will perform certain assigned functions for the system. Their total quantity is dependent upon the system application.

Each of the multi-processors specializes in one or more tasks, and each has its own program and data memory. Processors interact with each other via the CPU, or over direct data and address buses under a defined protocol.

Since the functional separation reduces real-time processing complexity, software design is an easier task in a system of this nature. CPU complexity can even be reduced as a result. In addition, this architecture permits a higher level of diagnostic processing to be incorporated, since time-sharing of functions in each processor becomes more tractable.

A clear disadvantage of the pure functionally-partitioned approach is that redundancy necessary for system availability may be very costly. If task takeover is used to reduce redundancy requirements, a more complex control system becomes necessary and software design also increases in complexity.

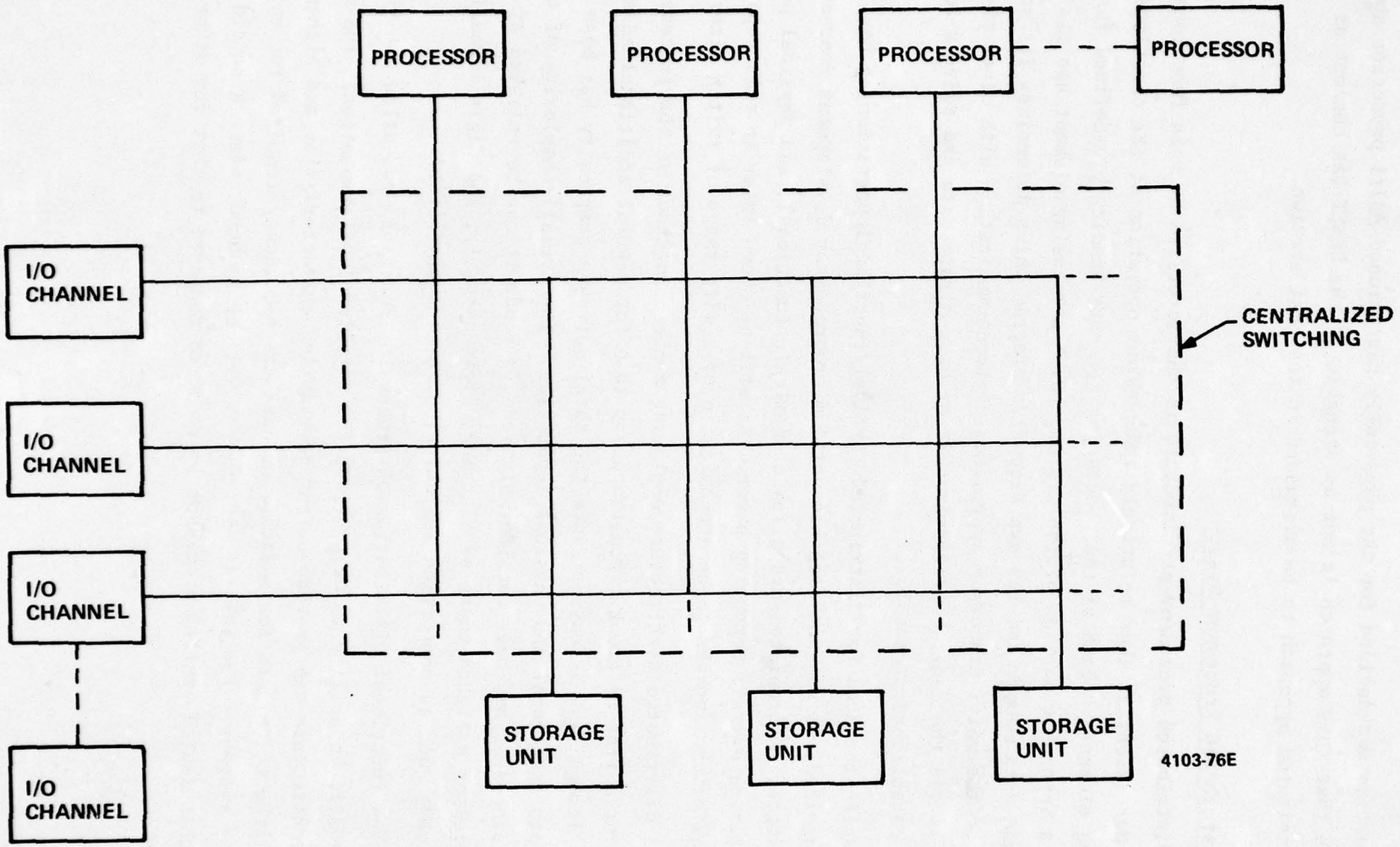


Figure 6-1. Polymorphic Processor Architecture

Just as was decided for the physically partitioned multi-processor approach, it appears that this approach is not as amenable to the SENET-DAX concept as the fully distributed approach to be discussed in the next section.

6.4.3 Distributed Processor System

Distributed processor architecture is characterized by both functional and physical partitioning, thus relatively independent operation of the constituent processing elements. Each of the processing elements handles pre-defined functions for only a separable portion of the system. Each processing element has its own program and data memory and its own executive program which determines its scheduling. Just as for the multi-processor approaches, intercommunication with other processing elements is via the central processor unit or over direct data and address busses under an established protocol.

While the need for distributed physical/functional partitioning has long been recognized, it was not feasible until microprocessor development reached its present state. Microprocessors allow economical, functional, and physical partitioning with increased overall computing power. As will be shown later in this report, this architecture appears to be economical over a wide range of switch sizes.

A distributed architecture employing a microprocessor as the processing element for a line or trunk circuit appears to offer several additional advantages. Software design task is easier since the total software complexity has been reduced due to both functional and physical separation. The overall complexity of the central processor has been considerably reduced since it can be relieved of such microprocessor assigned tasks as scanning, digit assembly, and digit signalling. Other tasks, such as connection switching, have been simplified.

The functional distribution of processing functions can allow a higher level of diagnostic capability to be provided for detailed fault isolation. The distributed architecture can provide better degradation characteristics and higher reliability without the need for software or hardware redundancy required for multi-processor systems. The initial system cost will be reduced, since a considerably less sophisticated central processor is required compared to that for other configurations.

For the reasons given, a distributed architecture based on micro-processors has been chosen for further consideration. This will be elaborated upon in the hardware and software discussion to follow.

6.5 EVALUATION OF SWITCHING MATRIX APPROACHES

6.5.1 Introduction

Addressed in this section is the comparison of various switching matrix approaches to determine whether traditional time and space division approaches are as applicable to the SENET-DAX concept as an entire processor approach as just described. Specifically, we will investigate and trade-off digital space division switching using logic gates; digital time division switching using the TTC-39 approach; and a distributed processor architecture.

As an objective, the switching matrix technique must switch digitized voice and data with a wide and continuous range of bit rates up to at least a 200-Kb/s channel rate, and a 1.544-Mb/s trunk rate. In addition, the matrix structure must allow for modular growth of the switching system, and must provide cost-effective implementation of the switching concept.

Three assumptions have made for this analysis: the SENET-DAX system concept is to be implemented; the switch must be non-blocking for Class II traffic; and the switch must service a maximum of 256-to 512-subscriber and 14 trunk terminations.

6.5.2 Digital Space Division Switching

6.5.2.1 Characteristics

In a digital network, digital space division switching is in essence logic gate switching. It has all the advantages of digital monolithic solid state technology and the flexibility of the wideband space division switching without the constraints and complexities of the analog crosspoint. A digital space division switch can transfer signals of several megabits per second through the switch and, therefore, would be applicable to the SENET-DAX concept.

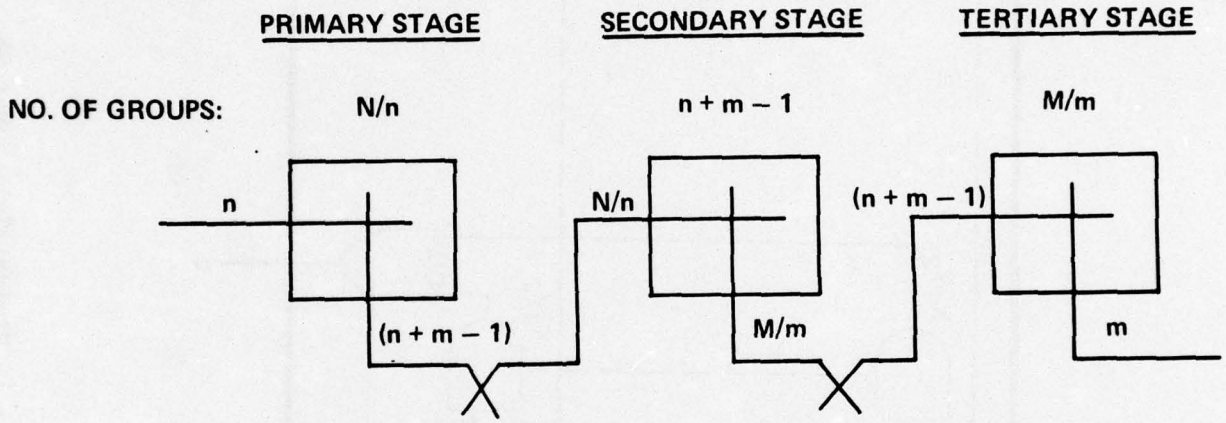
A variety of matrix designs is available for implementing the SENET-DAX switch. However, for the purpose of illustrating the digital space division switch, a three stage matrix shown in Figure 6-2 has been selected. It can be shown by well-understood methods (9) that a three stage matrix provides nonblocking switching when the number of center groups equals (n + m - 1). Thus, a 256-line nonblocking switch can use 8 inlets and 16 outlets per group for a total of 22,000 crosspoints or appriximately 90 crosspoints per line. A logic schematic of a typical 8 x 16 matrix group is shown in Figure 6-3.

A digital space division switch is implemented using logic gates and flip flops for switching and is, therefore, highly amenable to large scale integration. To illustrate the cost effectiveness of this matrix implementation, the 8 by 16 matrix could be placed in a 40-pin LSI package. One hundred ninety-two of these LSI packages plus a small amount of associated addressing circuitry could implement a 256-line matrix.

6.5.2.2 Evaluation

The digital space-division matrix is attractive from the stand point of hardware simplicity and LSI implementation. However, it is not suitable for the SENET-DAX concept because of two major disadvantages: the space-division matrix cannot perform the multiplex-demultiplex function which must, therefore, be implemented separately; and the varying size and time position of the DAX channel samples/packets would produce inordinately complex timing and control and require buffering not present in the space division matrix.

In future development, however, consideration should be given to the digital space-division matrix working in conjunction with other sub-systems such as microprocessors or time-division multiplexer units. This configuration could be used to control connections among subscribers local to the same switching central, or to connect these subscribers to the flexible time-division portion of the switch for access to SENET-DAX network trunks.



N = TOTAL NUMBER OF INLETS
M = TOTAL NUMBER OF OUTLETS
n = NUMBER OF INLETS PER PRIMARY GROUP
m = NUMBER OF OUTLETS PER SECONDARY GROUP 4109-76E

Figure 6-2. Digital Space Division Matrix

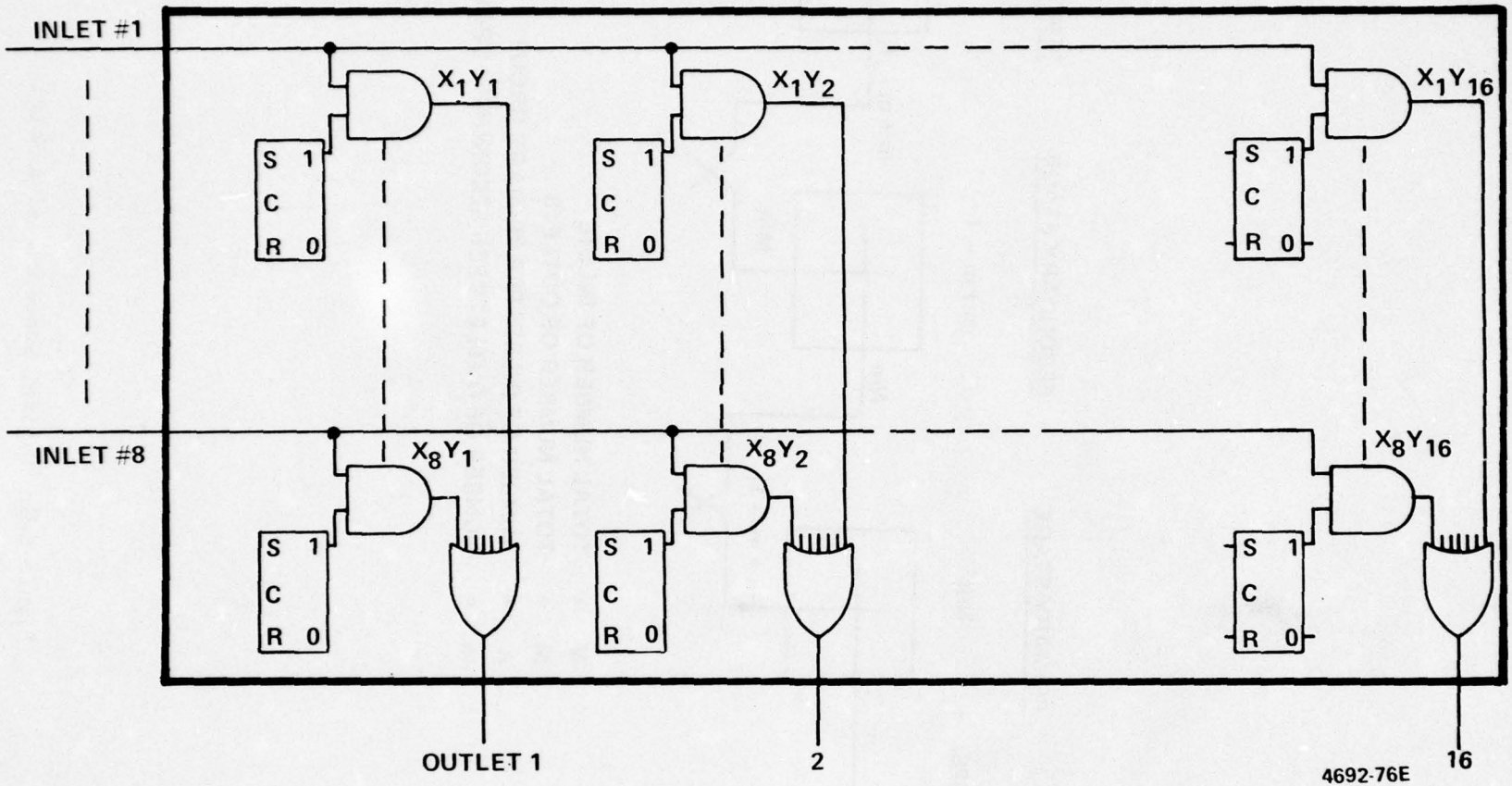


Figure 6-3. 8 x 16 Digital Switch Matrix

6.5.3 Digital Time Division Switching

6.5.3.1 Characteristics

The digital time-division switch approach considered is based upon the AN/TTC-39 design approach. Shown in Figure 6-4 is a digital time-division switch designed for a maximum data channel rate of 256 Kb/sec. Each input, regardless of its input data rate, is sampled at 256 K samples per second.

In this configuration, a switch memory word is provided for each switched channel. The memory word, which contains both a data section and an address section, accepts incoming data bits in the order of their arrival and reads out the bits in a different order based upon the connection address stored in the address section of the word. The digital time division matrix operates in a synchronous manner with a fixed channel bit rate. Binary submultiples of the highest channel bit rate can be synchronously multisampled, i.e., an exact number of repetitions of each bit is provided. Asynchronous or non-synchronous data can be multisampled with a requirement that the switch channel bit rate be approximately an order of magnitude higher than that of the channel being switched. However, the multisampling of lower data rate channels results in a lower multiplex efficiency and in an increased memory requirement for a given throughput capability.

6.5.3.2 Evaluation

The digital time division switch approach is an excellent approach for a system which operates in a synchronous digital network with fixed channel bit rates and with channel and subchannel bit rates which are binary submultiples of the highest switch channel bit rate. However, the SENET-DAX concept addresses the switching of diverse channels with a wide range of data bit rates which, especially in the case of external networks, are neither submultiples of the maximum channel rate nor synchronized to a common or extremely high accuracy timing standard. The variable time slots of the SENET-DAX master frame structure, which contain varying numbers of bits depending upon the channel bit rate and the dynamic reallocation of these time slots, make the SENET-DAX switching concept incompatible with the fixed channel characteristic of the digital time division matrix. The digital time division approach is, therefore, not recommended for implementing the SENET-DAX concept.

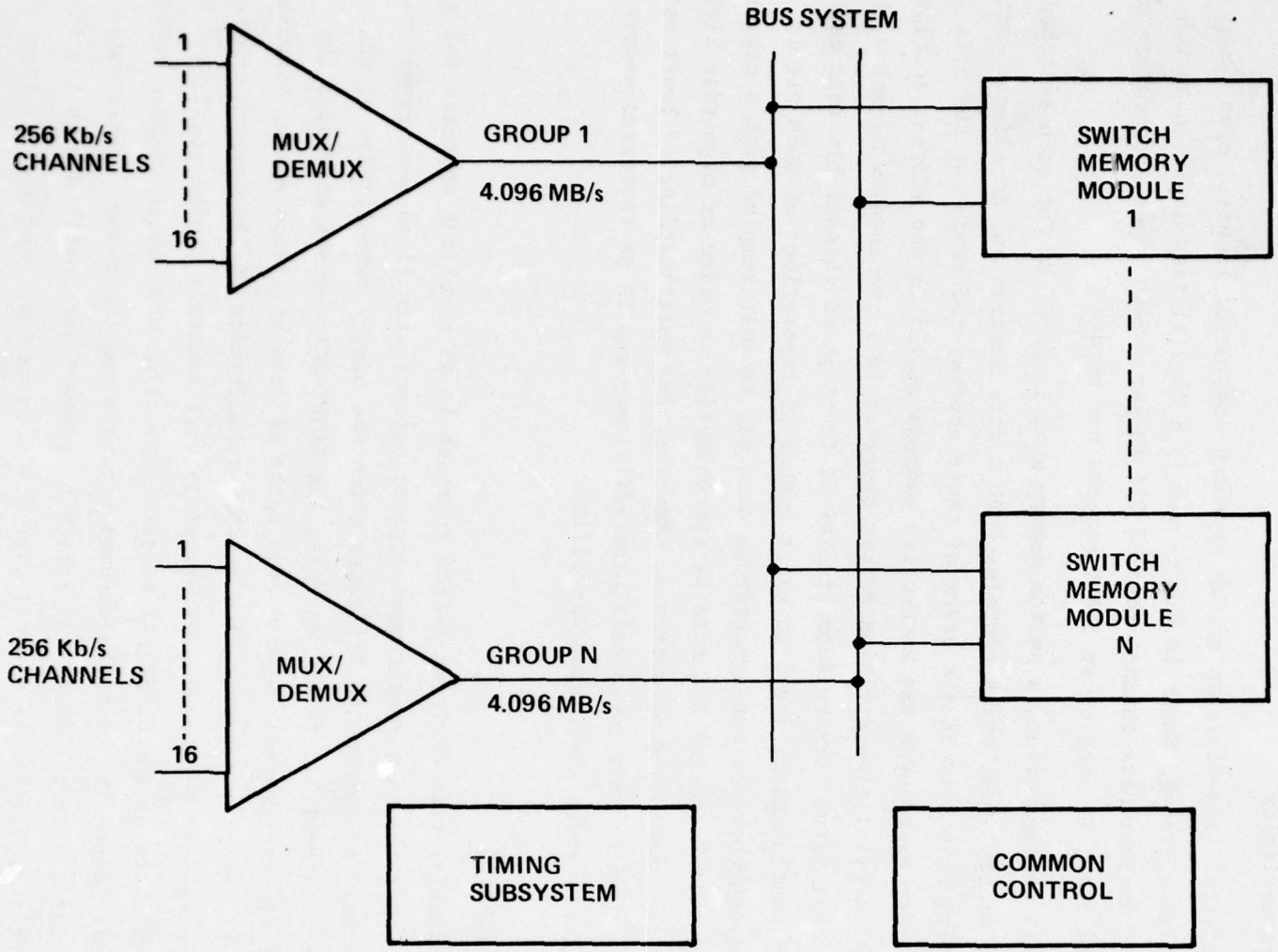


Figure 6-4. Digital Time Division Switch

6.5.4 Microprocessor Time Division Switching

6.5.4.1 Characteristics

The microprocessor time division switch approach illustrated in Figure 6-5 uses a distributed buffer memory to perform the slot interchange switching function. A group memory and a microprocessor control are associated with each input-output group. The group memory stores the incoming data of the associated groups' master frame(s) under the control of the microprocessor(s). The microprocessor(s) also accesses output data from its associated group memory or from other group memories via the bus system for output. An overflow memory is available to all group memories to alleviate overflow conditions.

The common control is provided to perform common switch functions such as signalling and routing. The microprocessor time division switch uses data processing techniques to access and control switched data in memory and thus provide the flexibility required for switching variable format time division multiplexed data.

6.5.4.2 Evaluation

The microprocessor time division switch appears to be an excellent approach for the SENET-DAX switching system concept. The microporcessor switch provides the data processing control techniques which are best suited for handling the variable time slots of the SENET-DAX master frame structure. The 10-msec master frame cycle of the SENET-DAX switch concept coincides well with the timing of a microprocessor data system. The addressing flexibility of the microprocessors can provide the dynamic reallocation of time slots which permits the SENET-DAX switch's realization of increased trunking utilization. The distributed memory and control provide a cost-effective switching system by minimizing the amount of common control which must be duplicated to achieve a given level of system availability. The microprocessor time division switch approach is, therefore, recommended for the architecture of the SENET-DAX switch. It is described in detail in Section 8 of this report.

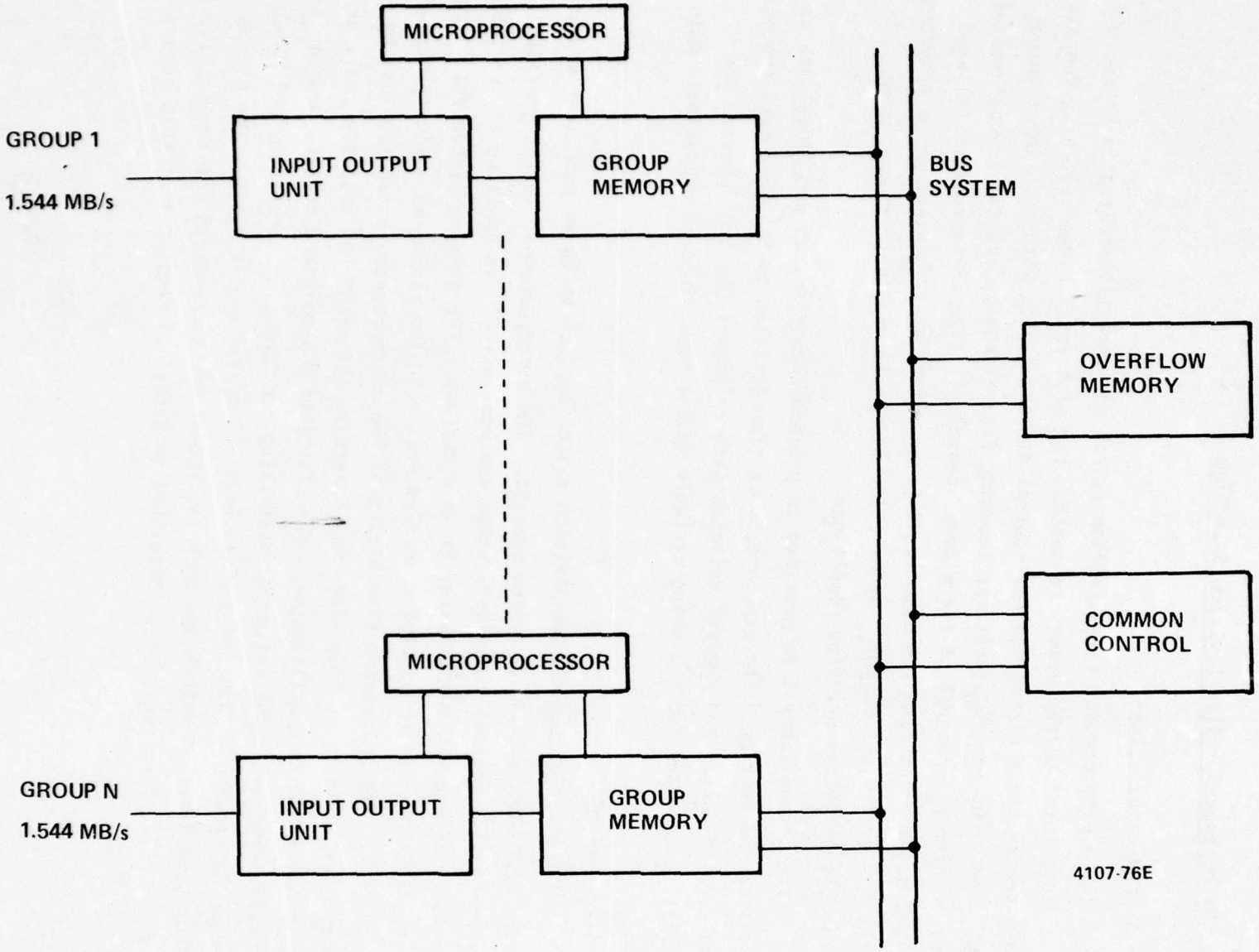


Figure 6-5. Microprocessor Time Division Switch

SECTION 7

SYSTEM SOFTWARE ARCHITECTURE

SECTION 7

SYSTEM SOFTWARE ARCHITECTURE

7.1 PROBLEM

Given the distributed architecture recommended for the SENET-DAX system, as discussed in Section 5, and the processing requirements that were examined in prior sections, we wish to identify and outline a software structure that will be responsibe to the needs of the system. Specifically we wish to determine the processing procedures imposed by a constant period, self-synchronizing frame containing digitized "circuit-switched" and data information transmitted synchronously at T1 carrier rates and implemented by multiple distributed processing elements.

7.2 OBJECTIVES

The objectives of this investigation are similar to those for processing architecture tradeoffs, since the determination of processor and software structure are integrated efforts. These objectives are flexibility for dynamic allocation and for extension to higher transmission rates; ability to incorporate a high level of diagnostic capability; maximum advantage of advanced hardware technology where this reduces software complexity, is economical, and meets system goals; and a contribution towards the objectives of reliability, modularity, and expandability of the system. More specific objectives for software architecture are to provide a process for reception of real time data and control of direct memory access storage, and to relax the processing time requirements of the port or link processes by distribution of the processing functions over the distributed processing elements.

7.3 PROGRESS

7.3.1 System Overview

Figure 7-1 shows a Digital Access Exchange functional structure of four major components. These are the access processes, the voice/packet stages, the network interface, and the SENET Network Processor (SNP), with the SNP providing all required network communication functions. This division permits complete insulation of the DAX access processes from the backbone network and allows continued operation of network functions in the event of an outage in the access function. In order to maintain a clear view and therefore provide an efficient design for those attributes

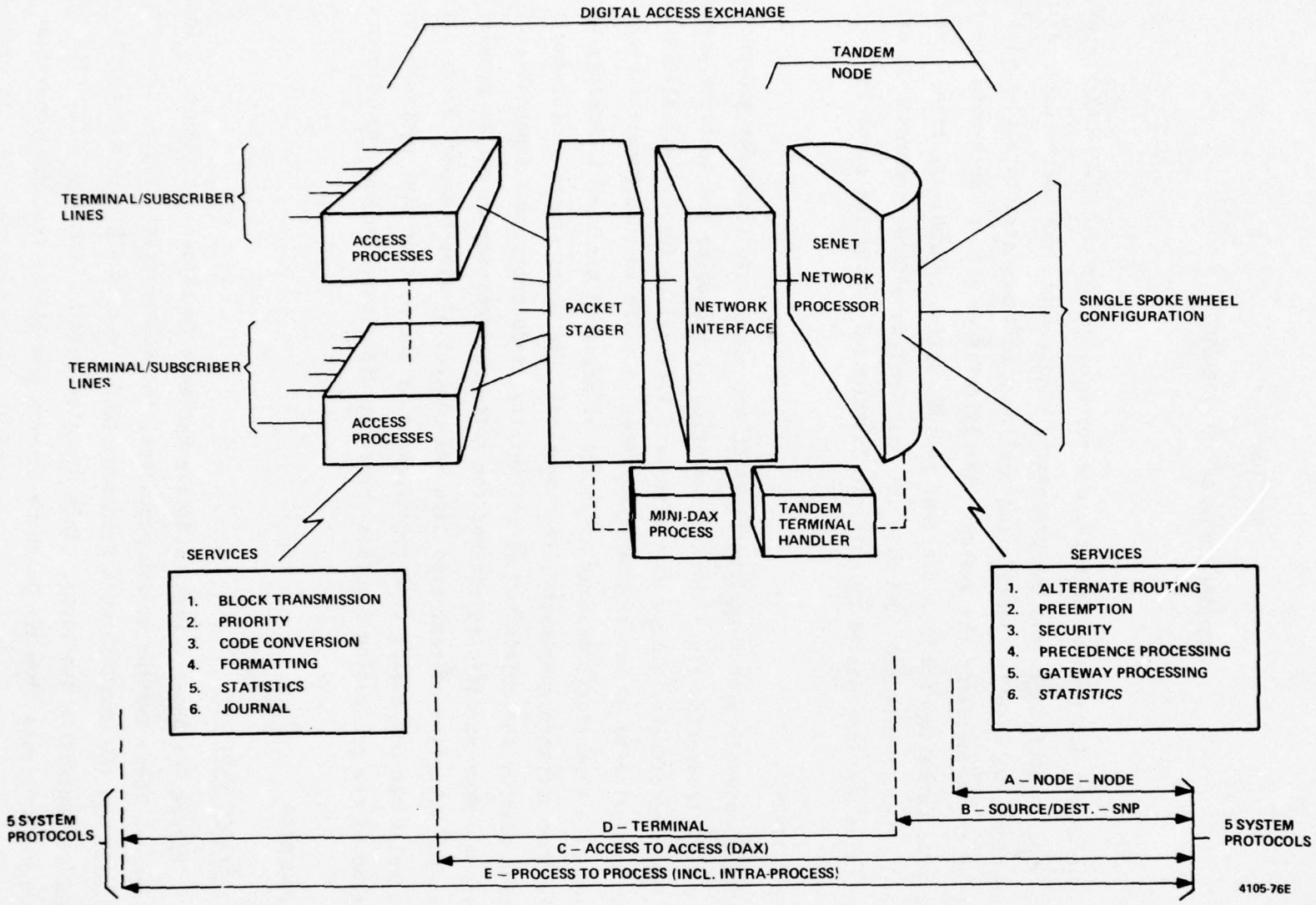


Figure 7-1. Digital Access Exchange (DAX)

essential only to the SENET-DAX concept (and not only to message or packet switching in general), those features considered to be service-oriented are identified in separate blocks. This calls attention to their fundamental nature as modular additions to the basic concept.

The SNP has been conceptually provided with a direct input/output peripheral interface (Tandem Terminal Handler), which permits a portable or permanent console to be connected for diagnostic or program administrative purposes. If communications with another network node and desired via this interface, then there must also be an optional software interface (Mini-DAX process) into the packeting module as shown by the dotted line connection in Figure 7-1.

For subsequent discussion, five system communications protocols are shown at the bottom of Figure 7-1. The classifications are consistent with standardization efforts performed by other industry study groups and are, for the most part, self explanatory. E-level protocol is totally transparent to our system, since it is a device-to-device communication procedure. It has been included for the sake of completeness and to display its relationship to other protocol levels.

Review of alternative architectural considerations (Section 6) has led to a recommendation of a SENET-DAX processing structure that utilizes a distrubted multiprocessor architecture, where processors are configured on a "port" or link basis, and advantage is taken of the existing need for multiple links by providing redundant processing elements.

Operating asynchronously, data transmission/reception and processing are permitted on all port processors at the same time, thus relaxing the speed requirement. This high degree of simultaneity is identifiable as a "polymerous" process, a process that exhibits not only a high degree of parallel processing, but also one in which that processing is interrelated in such a manner that each output is convolved in parallel with all other outputs as a result of simultaneous processing of all inputs. Cross switching is achieved by polymerous processing of the data over a redundant tri-level data bus resulting in a system with polymorphic-like qualities but without much of the expense for sophisticated switchover circuits and their control. This arrangement reduces the processing requirement of those functions that are common or central to the node. The common types of functions, which include system constants and routing table processing, are assigned to a nodel processor that also operates in a background mode for processing lessor or "non-essential" services such as traffic data collection, debugging and device input/out-

put. Functional minimization in this area is expected to reflect itself in a significant reduction in required hardware redundancy for the nodel processor. Refer to Figure 7-2 for a block diagram of the architecture.

7.3.2 System Software Structure

7.3.2.1 Link Processing Structure

The link architecture utilizes a structure which is intended to insulate the link input processor from real time data processing constraints. This organization is also true of the output link processor. The processing elements responsible for real time processes are referred to as a link synchronizer and an input link and output link sequencer. Only one synchronizer is required on a per link basis as this element is shared by input and output for Start-of-Frame (SOF) detection, generation and frame correllation.

As a result of the unique architecture and the intimate sharing of processing functions by both hardware and software, conventional methods of description do not seem to be adequate. While a certain degree of background/foreground processing can be identified in a particular processing element, it does not hold true for the entire node. Since all elements of all links are simultaneously active and an interrupt structure is imposed to effect an intra-link data control as well as an inter-link information transfer, a high degree of true nodal multi-processing is achieved on a data driven distributed processing structure.

To further develop the concept of the link processing structure, the pictorial diagram of Figure 7-3 was developed to show the relationship of the processing flow on one link to the incoming constant frame period as well as the functional distribution of these processes over the link processing elements. Figure 7-4 is the corresponding diagram relevant to the flow of link output processing.

The processing elements named in the above diagrams essentially comprise the modular link-oriented structure and a conceptual representation of these distributed processing elements is shown in Figure 7-5. This should be examined in concert with Figure 7-2 and the processing flow diagrams of Figures 7-3 and 7-4, and be referenced in the susceeding discussion.

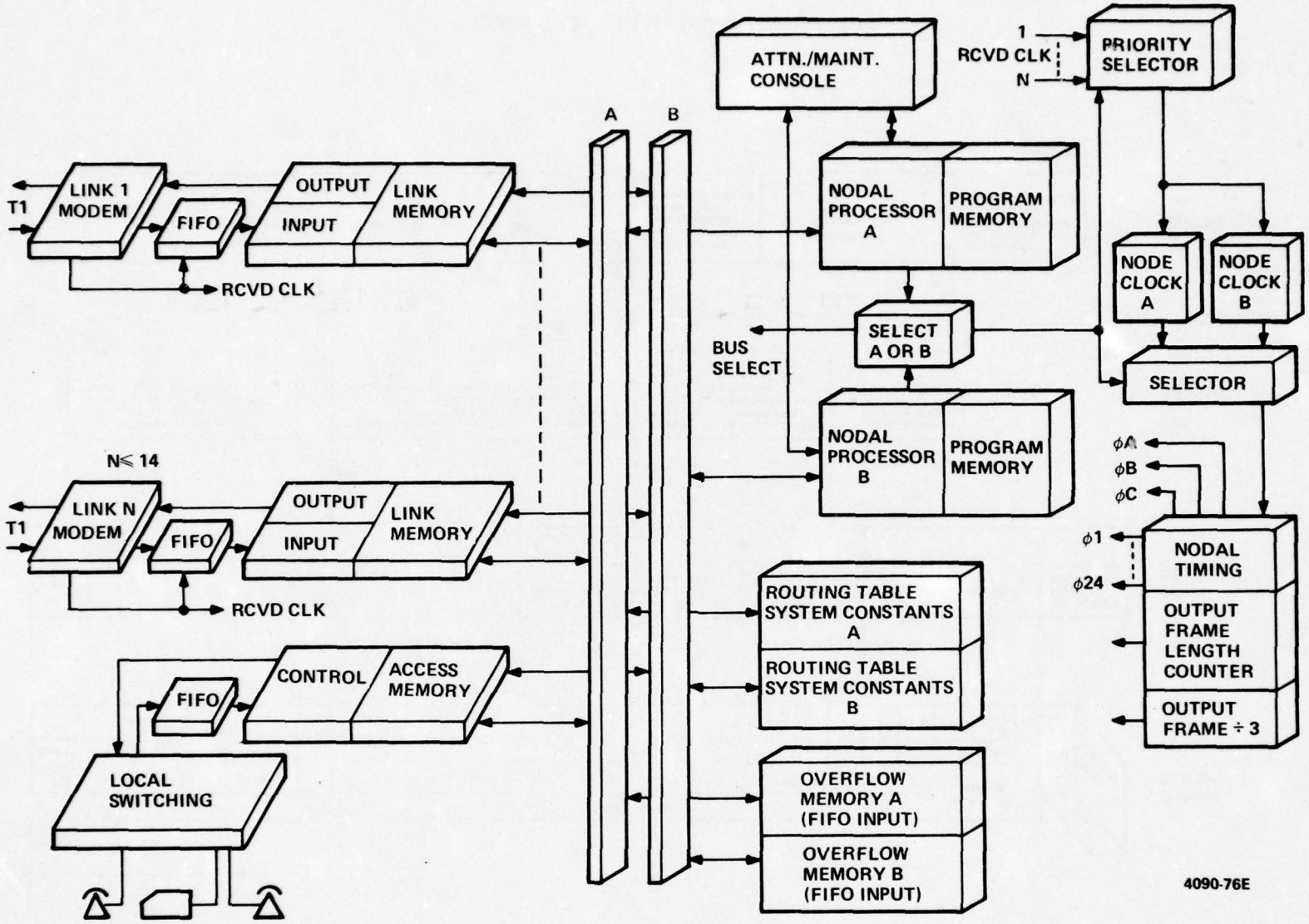


Figure 7-2. Node Processor Architecture

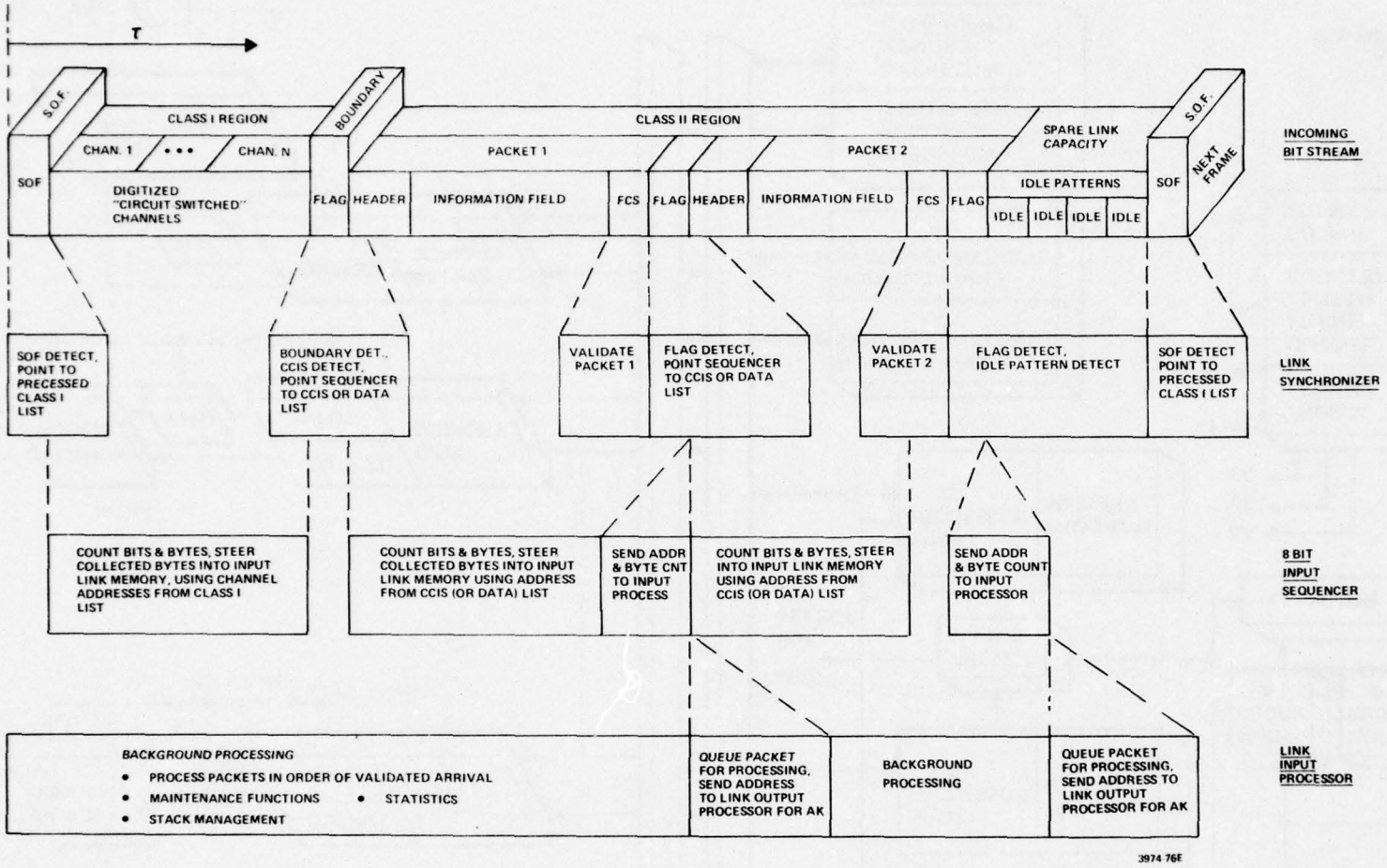


Figure 7-3. Link Input Processing

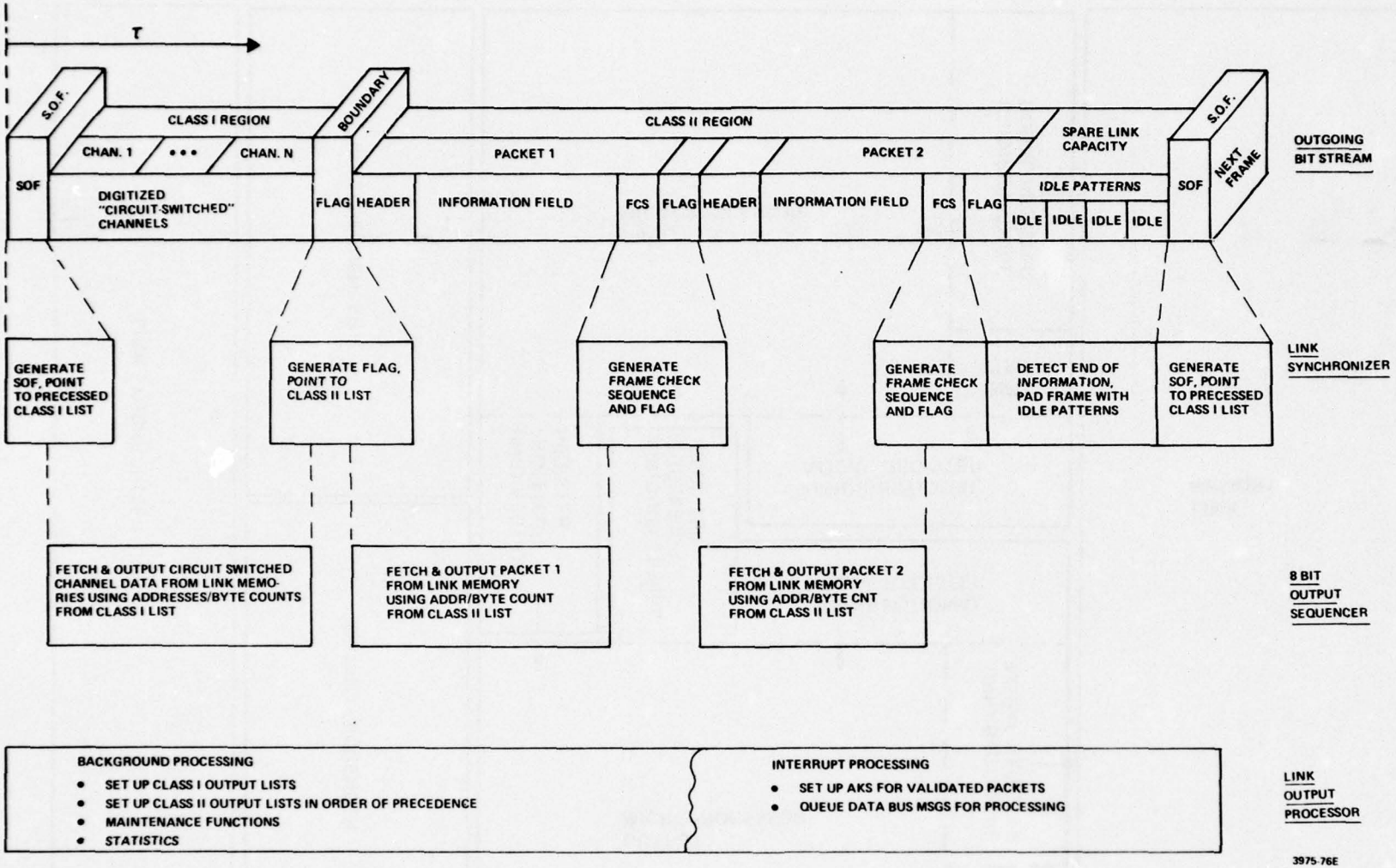


Figure 7-4. Link Output Processing

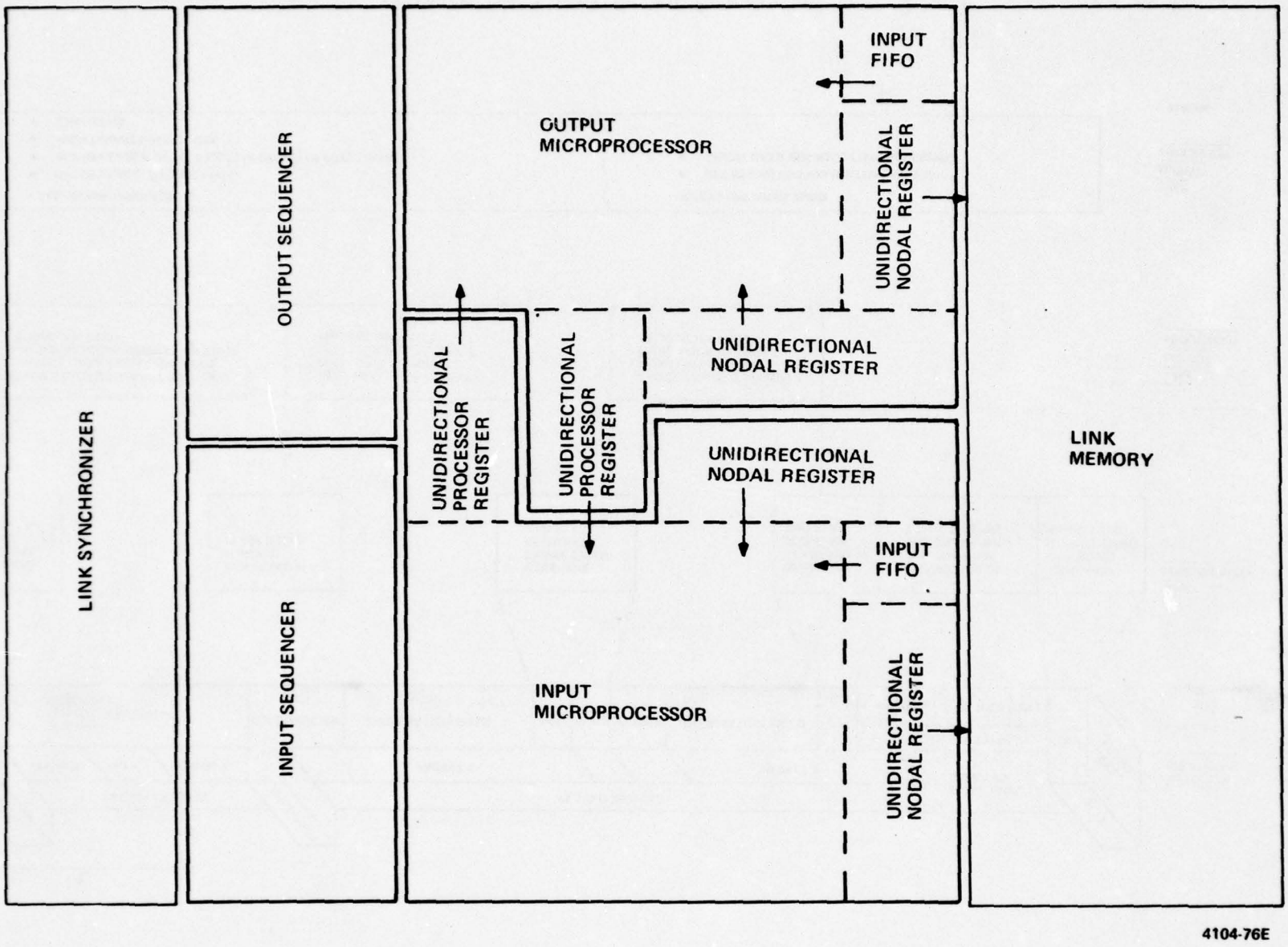


Figure 7-5. Link Processing Structure

The link processing elements shown in Figure 7-5 work cooperatively with all other link elements, via the data bus, to accomplish the switching function. Time multiplexing/demultiplexing is provided by the link output and input processing. The flow of data through this process is shown in Figure 7-6.

Directed control of the voice/data input is achieved by forwarding the input storage memory address to the appropriate output link processor via the tri-level data bus upon completion of the input processing. As may be seen in Figure 7-6, each output processor directs the extraction of stored information by its own link output sequencer by providing its sequencer with the processed linked list addresses upon completion of the output processing.

7.3.2.2 Frame Processing Procedures

7.3.2.2.1 Link Synchronizer - The link synchronizer performs the following functions:

a. Frame correlation
b. Start-of-Frame (SOF) generation
c. Class I/II, delimiter detection
d. Class II flag detection/generation
e. FCS validation/generation
f. Out-of-synch recovery
g. FIFO Status Monitoring.

Implementation of the above functions by the link synchronizer relieves much of the real time frame-oriented processing from both the input and output microprocessors. Acquisition and maintenance of frame synchronization could be implemented, as an alternative, by software. This would permit flexbility in the assignment of the SOF pattern(s) and could be varied on a link basis. Two conditions may be monitored by the nodal processor for status; an "in-synch" condition (F1) and a "Remote DAX out-of-synch" condition (F4). If the frame maintenance and acquisition algorithms shown in Figure 7-7 are implemented in firmware it would be possible to vary the correlation threshold values on a per link basis and thereby particularize any given link for its own noise environment.

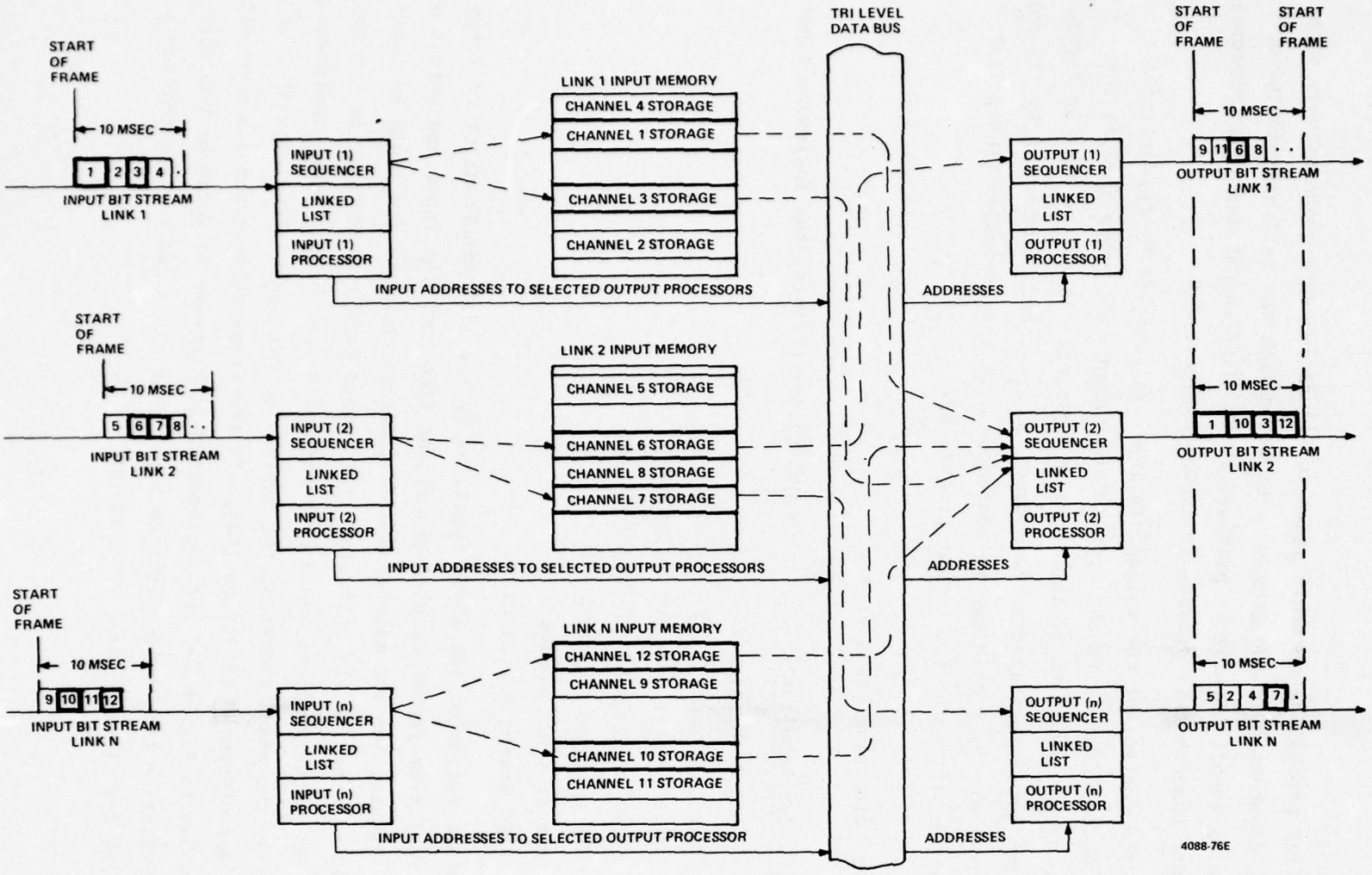


Figure 7-6. DAX Input/Output Directed Control

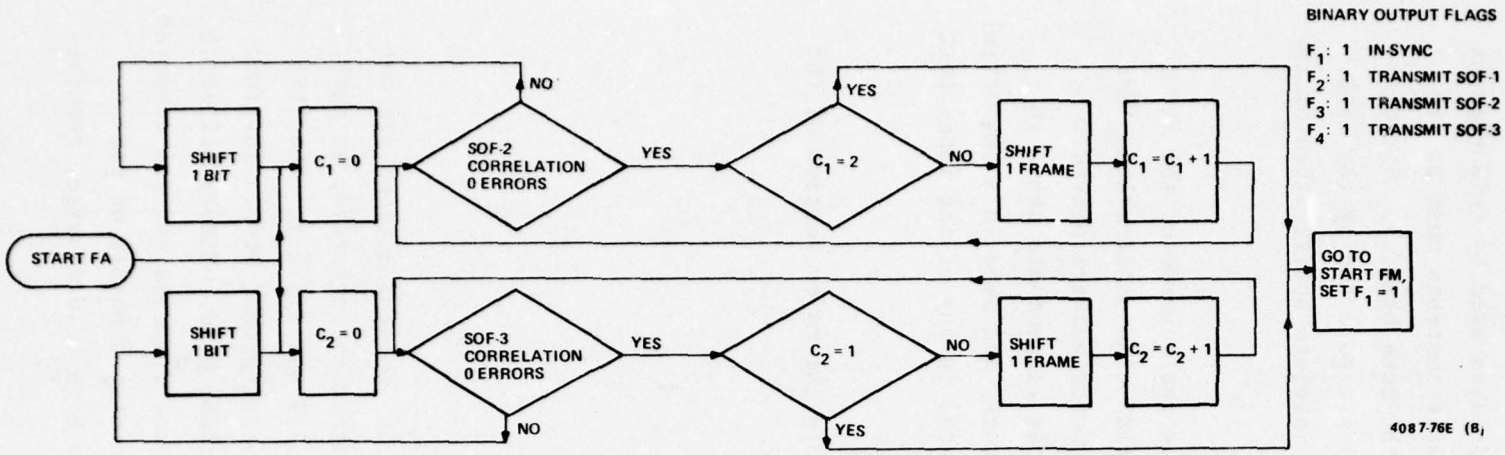


Figure 7-7a. Frame Acquisition Algorithm

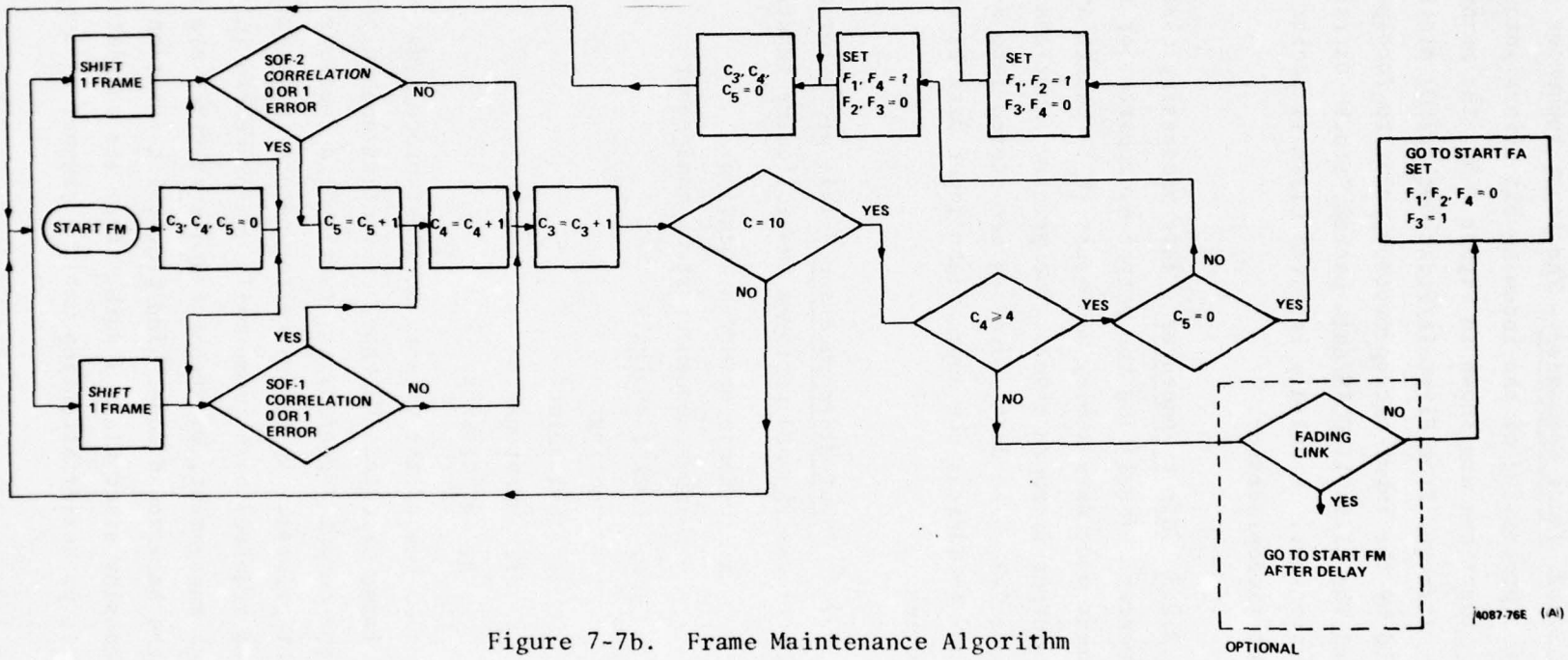


Figure 7-7b. Frame Maintenance Algorithm

7.3.2.2.2 Input Sequencer - The input sequencer provides real time Direct Memory Access processing of the incoming bit stream into the input port memory. The primary functions are shown in Figure 7-3. The memory addresses used by this element are extracted from First-In/First-Out (FIFO) stacks, whose contents must be provided by the input microprocessor prior to receipt of the data byte(s). The Class I and The Class II CCIS/data packet "stack" extractions are governed by the link synchronizer. A validly received Class II packet will initiate notification of the input microprocessor.

7.3.2.2.3 Output Sequencer - This processing element is the inverse of the input sequencer. Insulating the output microporcessor from real time constraints, this element uses data memory addresses, byte counts and sequence numbers provided by the output microprocessor. The primary functions of this element are shown in Figure 7-4. The data addresses are stored into FIFO stacks and used in a sequential manner to address the appropriate input data block located in any of the input port memories.

7.3.2.2.4 Input Microprocessor - This processor controls the major portion of the input processing and performs the following functions:

a. Dynamic memory allocation
b. Input sequencer stack management
c. Packet Analysis
d. Routing
e. Switching
f. Maintenance
g. Statistics.

Due to the variety of input processing functions, these functions are not all frame oriented and those of routing and switching are also performed, in part, by the output processor as described in Sections 7.3.2.2.5 and 7.3.2.5. The link input processor is perceived to operate in a background/foreground mode. The functions required on a frame basis are processed in foreground mode. Memory allocation, stack management, maintenance and statistics are not frame oriented and are executed in the background mode. The processing functions are centered mainly on packet processing since Class I data, after its initial set up by a CCIS message, requires little processor attention until a channel is dropped.

7.3.2.2.4.1 Packet Analysis - Each incoming packet is analyzed by the Link Input Processor (LIP) of the link on which it arrives. The packet is identified as either a data packet or a CCIS packet. Data packets must be analyzed for purposes of routing (i.e., to which outgoing link should it be directed) and also to determine the precedence level (i.e., where shall it be placed in the transmission queue). Note that precedence level can also influence routing.

Analysis of CCIS packets is more complex. Some CCIS packets are purely quasi-associated, i.e., intended for another switch and those are treated as data packets. Some CCIS packets are intended for this switch and must be analyzed more thoroughly. This type of packet often requires the generation of one or more CCIS packets by the recipient.

As indicated in Figure 7-3, the LIP queues the packet for analysis (processing) following validation by the link synchronizer and notification of its arrival by the link input sequencer. The actual analysis is done at background level.

Input packet processing is performed in order of arrival on the link, which is equivalent to processing in order of precedence.

7.3.2.2.4.2 Establishing and Breaking Down of Class I Calls - Procedures for establishing and breaking down Class I calls have been defined previously. Receipt of a "call initiate" CCIS will establish an input memory address and byte count for the call. After determination of the route and, therefore, the appropriate output link port address, the input processor forwards the memory address, byte count and precedence level over the tri-level data bus to the assigned output link. A routing algorithm is proposed which allows flexible alternate routing at each DAX as a function of individual call characteristics. The table-driven structure facilitates on-line adjustments from the nodal processor.

Routing of Class I calls from a DAX is controlled by data contained in three types of tables in common memory:

a. Directory Tables
b. Route Sequence Table
c. Search Sequence Table.

These routing tables are structured to take advantage of the fact that most of the interswitch routing sequences in a large network are common to many of the dialed directory numbers. Therefore, the Route Sequence Table is set up to be indexed by the route sequence number contained in the Directory Table entry for the dialed number. The Search Sequence Tables provide additional flexibility. These tables define the order and manner in which the primary and alternate links contained in the Route Sequence Table are searched for available channel capacity.

The interaction of the routing tables is illustrated in Figure 7-8. The design allows for automatic, graceful degradation of routine and low precedence traffic as interswitch links are marked out of service either for maintenance or as a result of adverse transmission environment or even partial network destruction. Higher precedence traffic will not be degraded but might be directed to less desireable routes. Thus, the switch will maintain an alternate routing capability under deteriorating network configurations that is continuously adaptive, without resorting to discrete fallback routing plans.

7.3.2.2.4.3 Packet Acknowledgement - Link protocol requires that validated packets be acknowledged to the sender. The acknowledge function is accomplished by cooperative action between the link input and output processors (LIP, LOP) of the same link. This information flow is handled by the undirectional link register. Alternatively, it would be possible to use the data bus for forwarding this type of information. However, the unidirectional processor registers permit intra-link communication without placing an additional processing burden on the data bus structure.

When the LIP queues the packet for analysis, it also forwards the packet sequence number and precedence level to the LOP which then adds an acknowledgement packet to the appropriate (precedence level) transmission queue.

7.3.2.2.4.4 Routing of Class II Data Packets - Data packet routing is accomplished by the LIP through reference to the directory and routing tables stored in common memory. When the route has been determined, the LIP puts the following information on the data bus:

a. The output link port address
b. The link memory address of the packet

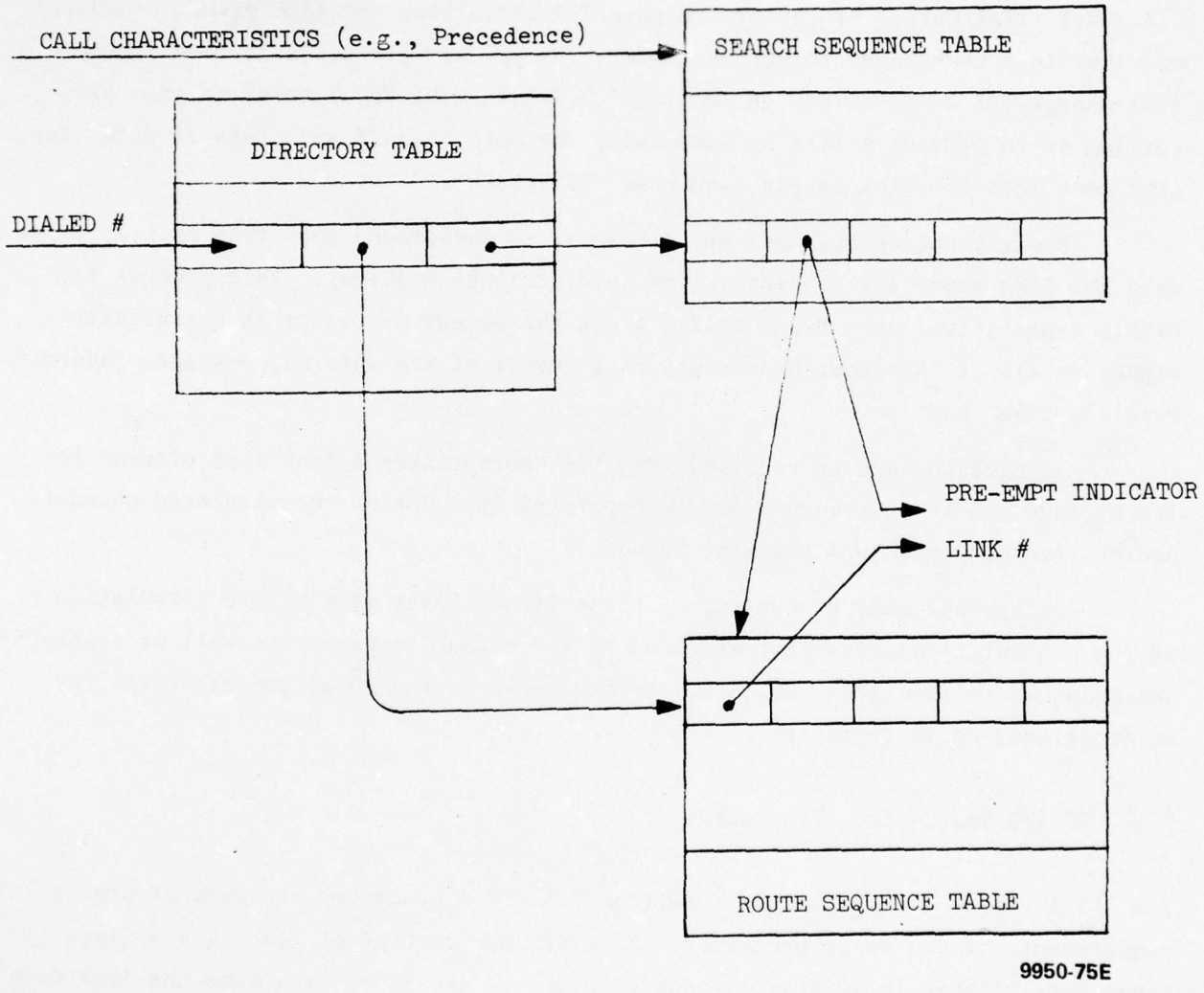


Figure 7-8. Flexible Deterministic Routing Table Structure

c. The byte count

d. The precedence level of the packet.

The appropriate LOP recognizes its port address and uses the remaining data to update its linked list as described in 7.3.2.5.3.

7.3.2.2.5 Link Output Processor - Figure 7-4 shows that the link output processor operates in a background/foreground mode. The primary processes involve linked list management as described in Section 7.3.2.5.1. The focal point of this processing is to provide a list of addresses, for both Class I and Class II data, for placement into the link output sequencer FIFO stacks.

Newly received messages are processed in foreground mode from both an input data bus FIFO stack and the intra-link unidirectional register. This process is highly repretitive, on a frame basis, since the output processor is essentially slaved to all of the input processors as a result of the internal messages (addresses) received from them.

As each message is received, the LOP appropriates a free list element for its storage and these elements are incorporated into one of the allocated channel, packet transmit or acknowledgement queues.

Background mode processing of these linked lists permits the formulation of the sequential address list required by the output sequencer as well as statistics and maintenance functions, which may be requested by the nodal processor via the unidirectional nodal registers.

7.3.2.3 I/O and Control Procedures

7.3.2.3.1 Class I Field Length - Section 7.3.2.2.4 describes the general processing requirements of the input processor. However, the control of Class I and Class II input data differs in several respects. Class I data is steered into the link data memory (by the input sequencer) by use of a starting address and a Class I field length count. The starting address may be fixed or administered as a program constant. Since triple buffering is employed, three starting addresses are required as hardware precesses the filling of each buffer (i.e., fills on an end-around basis). The length of the Class I field is passed to the link input processor from the link output processor via a unidirectional processor register for enabling the Class II flag detection function.

7.3.2.3.2 Link Input Connect Map - An input connect map, maintained by the input processor correlates the linkages established between input and output processors. For each "circuit-switched" channel the map contains the input memory address, byte count and output port address associated with that channel.

This map may be used by the maintenance routines for Class I input/output link error detection.

7.3.2.3.3 Channel Deletion - The processing burden of this function upon the input processor may be relieved by the use of a "broadcast" message to all output link processors.

Each output processor, upon recognizing its port address, would accept a channel delete request from the requesting input processor. Each output processor would determine if it contained an address from the requesting input processor. If an address were located, all subsequent higher addresses from the same requesting input processor would be adjusted by the byte count of the deleted channel. This processing is necessary if compression of the Class I information field is to be achieved.

Alternatively, the input processor would have to place a channel deletion request on the data bus for each channel subsequent in time to the channel to be deleted. This report assumes the former approach will be used to distribute the processing load.

7.3.2.3.4 Link Initialization - Both the input and output processors of all links manage free memory space by dynamically allocating fixed block size memory elements on an as-required basis. The output processors manage linked free lists consisting of 5 word list elements as described in Section 7.3.2.5.3. The initial linking of these elements into a list structure is done during system initialization under control of an administered pointer.

The input processors manage the dynamic allocation of free list elements in a similar manner except that allocation is done only for Class II CCIS and data packet storage. The address of each free list element is given to the input sequencer for placement in the FIFO stacks for subsequent input steering.

The initial linking of these free list elements into a list structure is done during system initialization under control of an administered pointer.

7.3.2.3.5 Block Size Estimation - Management of variable length CCIS and data packet storage represents a trade off between design complexity with increased processor load to achieve efficient memory utilization and design simplicity with decreased processor load (increased speed) with less efficient memory utilization. The latter approach has been proposed in this report, since the advantages appear to outweigh the inefficiencies when viewed in light of current hardware cost trends (reference Section 8).

Inefficient utilization of fixed block sizes due to variable length packets gives rise to the problem of optimizing the buffer size to minimize the expected storage requirements.

In the SENET-DAX system, a first-order approximation of the optimum block size is proposed which assumes a uniform distribution of message sizes between 1 and L. It can be shown that if each buffer contains M data words and A words of overhead, that the selection of M that minimizes expected storage is approximately $\sqrt{AL}$ (reference 13).

For linked list processing, three words of overhead are consumed by forward and backward linkages and a required input packet byte count. The maximum packet size is assumed to be nominally 2,000 bits which provides a data block size of:

$$M \approx \sqrt{3\ (250\ \text{words})} \approx 27\ \text{data words}$$

It is, therefore, proposed to fix the initial block size for Class II dynamic allocation of packet storage at thirty (30) words.

This result is considered to be extremely favorable for several reasons. First, analysis shows that the optimum block size is very close to AL for a great variety of distributions of message lengths. Second, and perhaps more importantly, if this block size is not exactly optimum for minimum storage it is correctly sized to contain the maximum length CCIS packet which also requires 27 data words. This means that control processing is simplified and need never be concerned with multiple-block linkages for CCIS packet processing.

7.3.2.3.6 Release of Free List Element - Initiation of the release of a list element, so that the input processor may restore the element of its free list (available to the input sequencer), must be done by the output processor assigned to the element upon its receipt of a proper acknowledgement from the LIP.

The memory address of the element is contained in the output linked list and this address is given to the input processor in whose link memory the data element resides.

The input processor may access the linkage pointers stored in the element for insertion into its free list or append the element to the end of this structure.

7.3.2.4 Nodal Processor

The nodal processor is responsible for functions related to the total switch and is insulated from direct involvement in real time input/output requirements. As such, this processor could conceivable operate in an in-line processing mode whereby each link would be serviced in a sequential manner. However, the approach recommended in this report allows a background/foreground mode of operation with link activity initiating service requests by interrupting the nodal background processing mode. Figure 7-9 is included to show the distribution of the operating systems over the various processing elements comprising the proposed architecture. This figure shows the processing relationship between the nodal processor and one of its links. Communication with each link is maintained thru unidirectional nodal registers (FIgure 7-5) which interrupt for nodal input-output processing.

Functions performed by the nodal processor include:

a. Diagnostics
b. I/O console/attendant interface
c. Debugging aids
d. Traffic data collection
e. Program (Link) administration
f. Routing/traffic.

It is expected that each link processor will contain a basic traffic data collection program that may be activated by command from the nodal processor. Due to the symmetry of the link memory, a specific function may be activated in all links by placement of an appropriate command in a fixed link memory location which is then

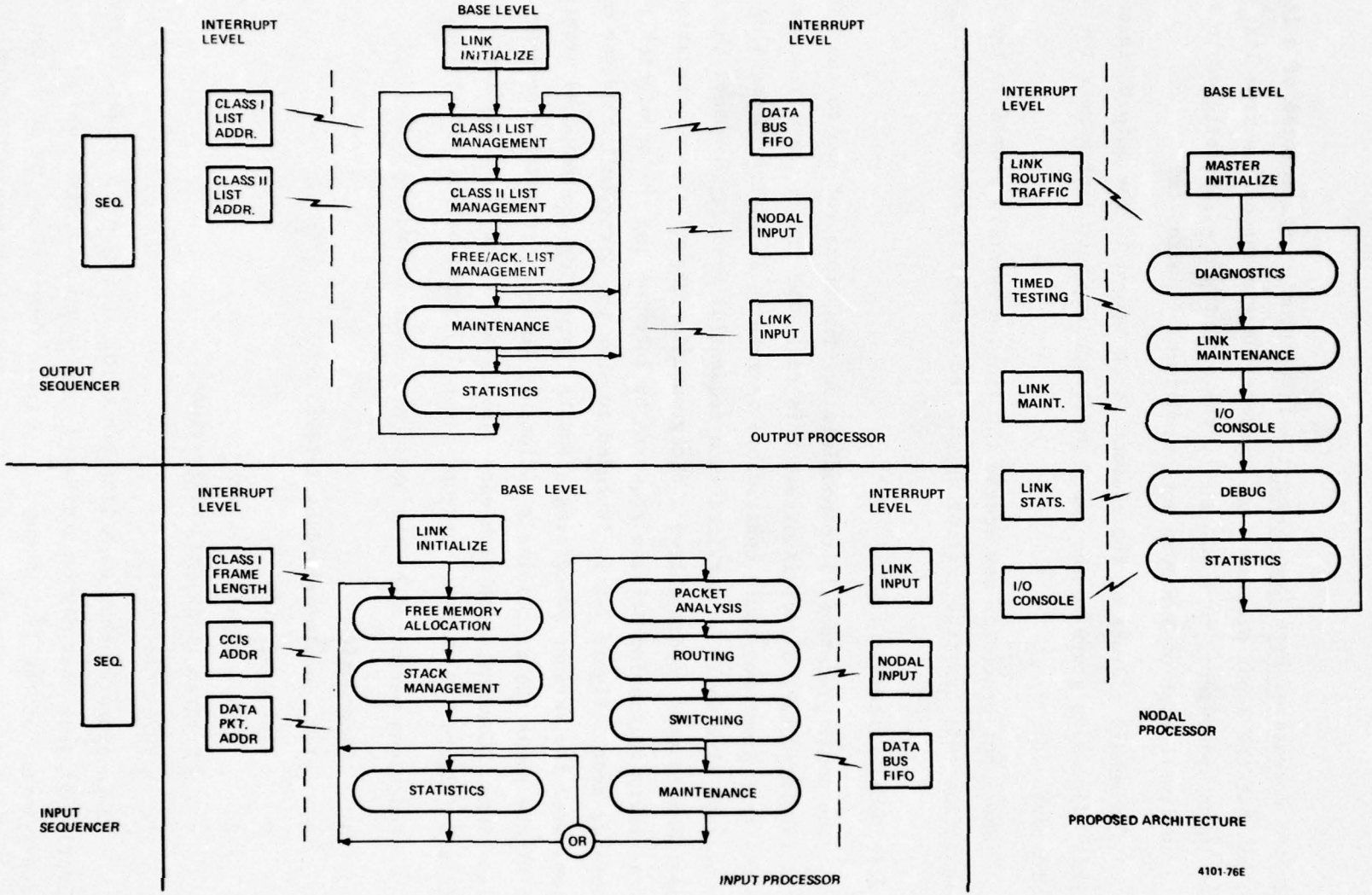


Figure 7-9. Nodal Processor

"broadcast" to all links by sequentially incrementing a 4-bit (high order) port address. The common location concept allows one universal program to be employed by all link processors as well as a simple method for link program administration by the I/O console/attendant interface.

Typical traffic data collection programs would include:

a. Snapshots
b. Timed summaries
c. Packet traffic
d. Retransmissions
e. Precedence usage
f. Preemption.

Route/traffic table updating is performed by the nodal processor as a result of reports issued to it by the link processors. Link activity data and/or channel capacity may be reported whenever the output linked lists or input channel maps are modified.

Special messages may be inserted onto any desired link, following composition by the nodal processor, as a result of nodal processing or via the console interface. Use of the "broadcast" facility for multiple link transmissions simplifies attendant console operational procedures while affording a flexible communications tool.

7.3.2.5 Switching and Multiplexing Procedures

Processing for switching and multiplexing functions of a node is distributed among the various processing elements of the node. The processing elements used to accomplish these functions are:

a. Link Input Processor (LIP)
b. Link Output Processor (LOP).

7.3.2.5.1 Linked List Management - As described elsewhere, the key to achieving the high data throughput proposed for DAX is the concept of managing a common memory element by manipulation of lists, rather than by physically moving the data. The use of linked lists makes feasible the idea of steering data into and out of non-contiguous memory blocks. The need for steering (or multiplexing) of output data is fairly obvious and a proposed linked list structure and procedures for managing that function is discussed below.

The need for input steering is not as firmly established. The use of triple buffering for Class I input data comprises input steering of a sort, but, the decision of whether or not to always gate the entire Class I region continguously into the appropriate buffer has not yet been made. The factors in the trade-off are basically port input processor loading vs. port output processor loading.

If the Class I buffer is always compacted, the dropping of a channel will result in a need for each output processor to adjust the addresses of all channels appearing later on the same incoming frame as the dropped channel. If the Class I buffer is not compacted, dropping a channel will create a "hole" in the input memory which can lead to a complicated and time-consuming input data memory management problem. The former choice (i.e., compacted Class I input data) is assumed in this report.

Although the Class I memory recycles itself automatically, with an entire input buffer becoming available for new input every frame, Class II input cannot be handled in the same manner. Input data steering is, therefore, proposed for Class II data, as discussed in Section 7.3.2.3.4 and below.

7.3.2.5.2 Input Linked Lists - Figure 7-10 shows a proposed structure for the Class II input linked lists used by the Port Input Processor to manage its memory resources as well as provide for data throughput.

In contrast to the output linked lists described in the next section, the input list elements describe a fixed length block of input memory with each element located within the block of memory it describes. In addition to the elements a pointer table is located in a fixed location of the input processor memory.

The size of the data blocks is anticipated to be 30 words including 3 words of control and 27 words of packet data (reference paragraph 7.3.2.3.5). Every block will be linked either to a free list (meaning available for storing input data) or a validated input packet list (meaning the packet is available for analysis or processing). A block will not be returned to the free list until the packet has been completely processed, and an acknowledgement received if transmittal to another switch is required.

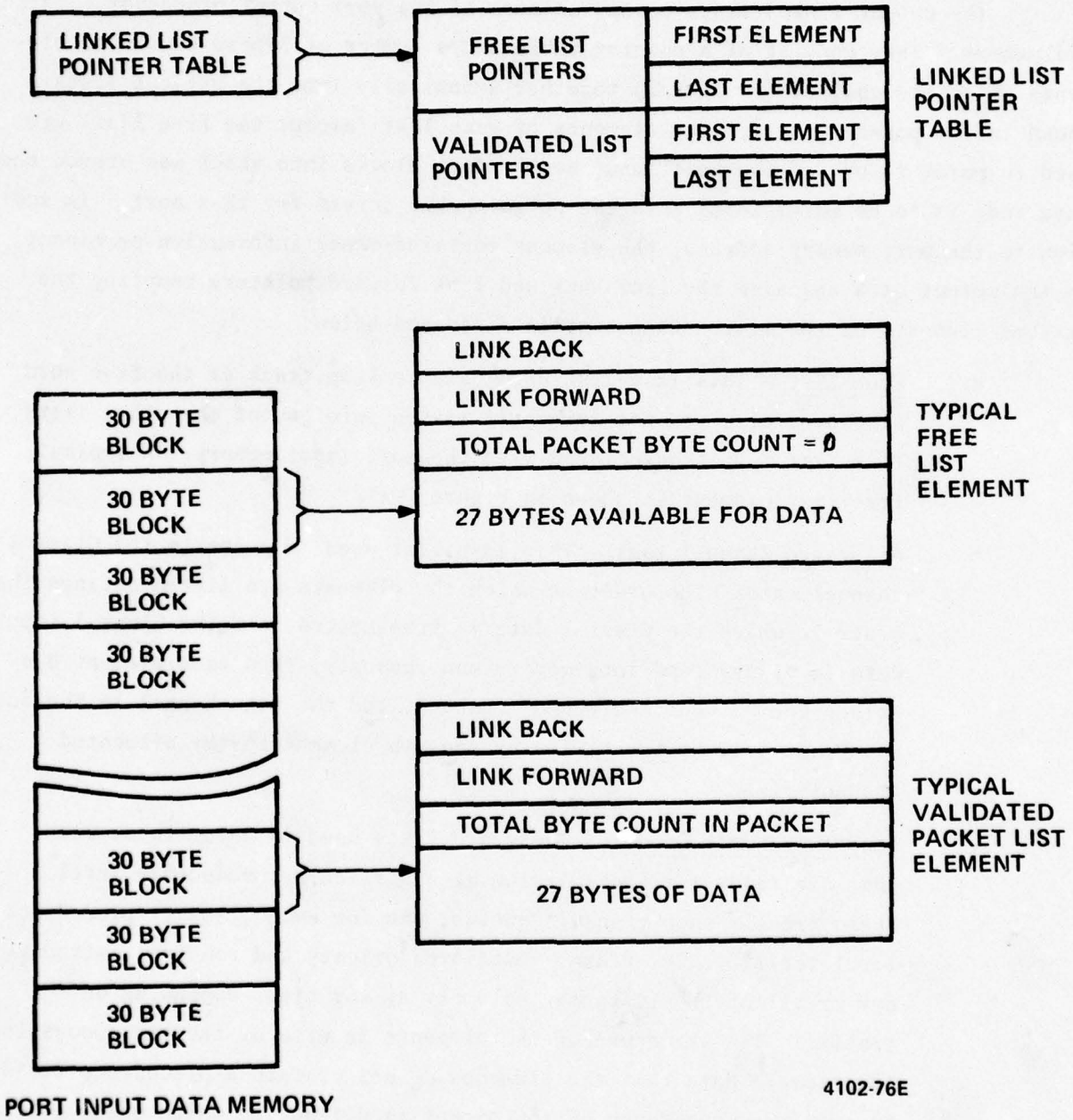


Figure 7-10. Port Input Processor Linked List

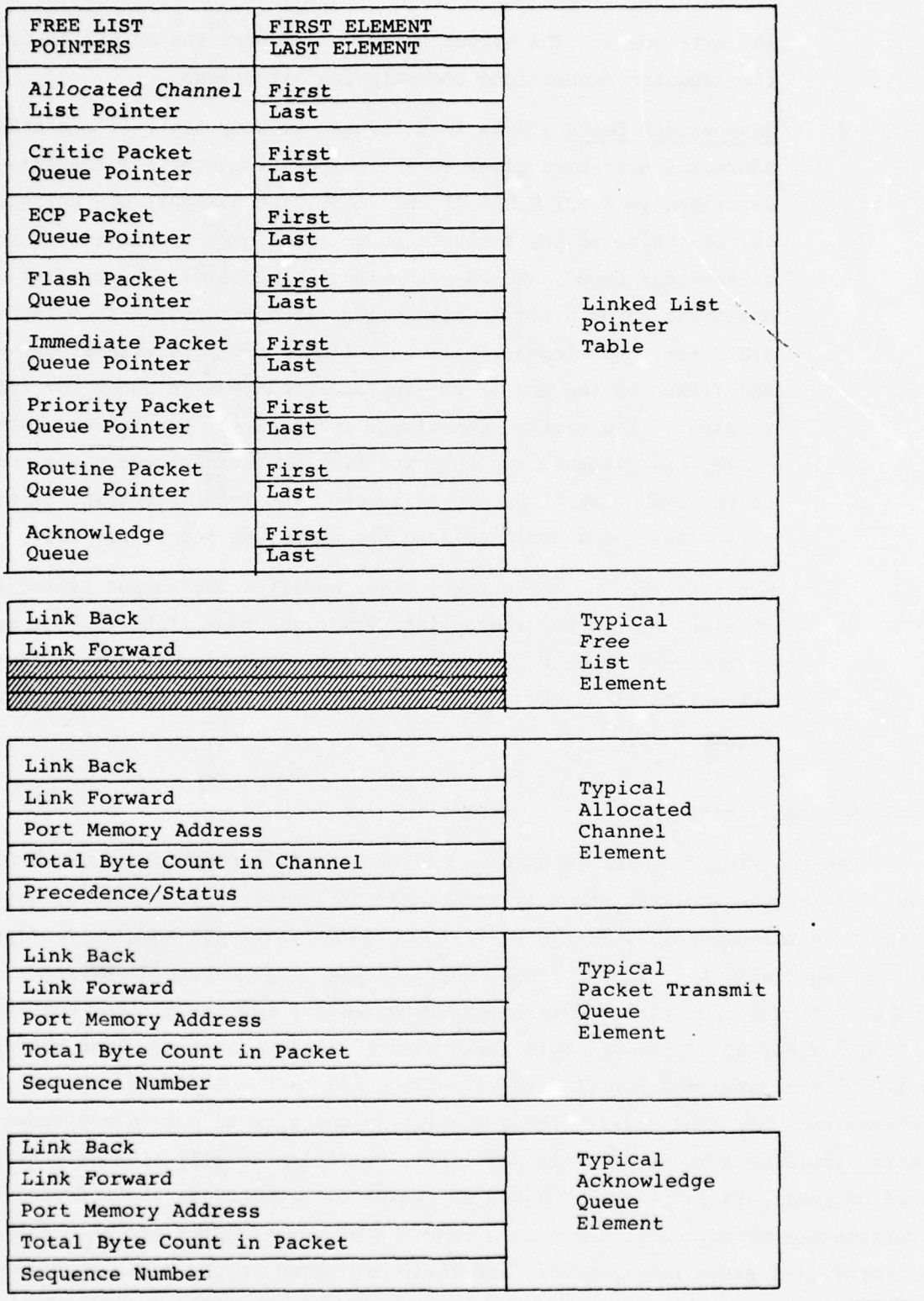
7.3.2.5.3 Output Linked Lists - Figure 7-12 shows a proposed structure for the linked lists used by the port output processor for managing the output multiplexing of both Class I and Class II data from the various port input memories.

The output linked lists occupy an area of the port output processor scratch-pad memory. They consist of a pointer table and a number of 5-word blocks or elements which are chained (or linked) together dynamically into the various lists shown in the pointer table. The elements of each list (except the free list) are used to point to one of the port input memory data blocks into which was stored the data that is to be multiplexed into the outgoing bit stream for this port. In addition to the port memory address, the element contains other information pertinent to the output data and also the link back and link forward pointers coupling the various elements of the list. Each list is discussed below.

a. Free List - This is a list used only to keep track of the five word blocks or elements not currently linked into one of the other lists. This list is not associated with the port input memory. A typical free list element is shown in Figure 7-11.

b. Allocated Channel List - This is a list used to point to the Class I channel data. The order in which the elements are linked defines the order in which the Class I data is transmitted. If the Class I input data is always read into memory contiguously, then each element describes one "circuit-switched" channel, and the ith channel in the output frame will be identified by the ith element in the allocated channel list.

c. Packet Transmit Queues - These are lists used to point to packets that are ready for transmission at a particular precedence level. There are six such transmit queues, one for each Class II precedence level (critic, ECP, flash, immediate, prioity and routine), although any or all of the lists may be empty at any time, depending on traffic. The structure of the elements in each of the six queues is identical. Note that the elements do not contain a precedence field because the precedence of the packet is defined by which queue it is linked into. The sequence number is established by the port output processor and sent to the output sequencer concurrently with the address



4695-76E

Figure 7-11. Port Output Processor Linked Lists

and byte count. The output sequencer handles the actual insertion of the sequence number into the outgoing bit stream.

d. Acknowledge Queue - This list is used to keep track of packets whose addresses have been given to the output sequencer for transmission as described in 7.3.2.2.5. At that time, the elements are unlinked from the beginning of the transmit queue and linked to the end of the acknowledge queue, and an acknowledge timeout will be initiated. If notification of a timely acknowledge is not received from the input processor, the elements will be unlinked from the acknowledge queue and linked to the end of the appropriate transmit queue for retransmission. If a timely acknowledge is received, two actions take place. First, the elements are unlinked from the acknowledge queue and linked to the end of the free list. Second, the input processor is notified to release the associated data blocks to the input free list.

Whenever the Port Output Processor modifies the output linked lists, it also updates the appropriate Traffic Summary Tables which are used by the Port Input Processors and Nodal Processor in the routing and preemption algorithms. The structures of these tables are shown in Figure 7-12.

7.3.3 Storage Timing

As the study results focused on the recommended architecture, is became clear that data storage would be strongly influenced by timing considerations, mostly due to the high through-put indicated by traffic requirements and network configuration. This is especially true in the tandem nodes of the singly-spoked network. Table 7-1 tabulates timing and storage data memory requirements for the various configurations up to a maximum of 14 links. This table assumes that a constant frame equivalent to 15,440 bits is utilized for T1 carrier service and that this frame input is triple buffered on a per link basis. These considerations give rise to a non-redundant memory requirement of 46,320 bits per link. Multiples of this storage requirement, based on configuration, are displayed as "words of storage" in the referenced table. Corresponding memory cycle times as a function of word lengths are also tabulated for the total link rates (bits/second) and these are shown in the three rightmost columns.

CLASS I TRAFFIC SUMMARY TABLE		
TRUNK	PRECEDENCE LEVEL	# OF BYTES
OUTGOING TRUNK 1	TOTAL CLASS I	ALLOCATED
		RESERVED
	R PRECEDENCE	ALLOCATED
		RESERVED
	P PRECEDENCE	ALLOCATED
		RESERVED
	I PRECEDENCE	ALLOCATED
		RESERVED
	F PRECEDENCE	ALLOCATED
		RESERVED
	F.O. PRECEDENCE	ALLOCATED
		RESERVED
OUTGOING TRUNK N	F PRECEDENCE	ALLOCATED
		RESERVED
	F.O. PRECEDENCE	ALLOCATED
		RESERVED

CLASS II TRAFFIC SUMMARY TABLE		
TRUNK	PRECEDENCE LEVEL	# OF BYTES
OUTGOING TRUNK 1	TOTAL CLASS II	TRANSMIT QUEUE
		ACKNOWLEDGE QUEUE
	R PRECEDENCE	TRANSMIT QUEUE
	P PRECEDENCE	TRANSMIT QUEUE
	O PRECEDENCE	TRANSMIT QUEUE
	Z PRECEDENCE	TRANSMIT QUEUE
	Y PRECEDENCE	TRANSMIT QUEUE
	W PRECEDENCE	TRANSMIT QUEUE
OUTGOING TRUNK N	TOTAL CLASS II	TRANSMIT QUEUE
		ACKNOWLEDGE QUEUE
	R PRECEDENCE	TRANSMIT QUEUE
	P PRECEDENCE	TRANSMIT QUEUE
	O PRECEDENCE	TRANSMIT QUEUE
	Z PRECEDENCE	TRANSMIT QUEUE
	Y PRECEDENCE	TRANSMIT QUEUE
	W PRECEDENCE	TRANSMIT QUEUE

4686-76E

Figure 7-12. Traffic Summary Tables

TABLE 7-1. STORAGE AND TIMING REQUIREMENTS -- DATA MEMORY

TYPE OF NODE	DEGREE OF REDUNDANT INPUT STORAGE	WORDS OF STORAGE*			I/O RATE - WORDS/SECOND MEMORY CYCLE TIME - MSEC./WORD		
		1 BIT/WORD	4 BITS/WORD	8 BITS/WORD	1 BIT/WD	4 BITS/WD	8 BITS/WD
5 LINK REGIONAL NODE	NON-REDUNDANT (Single Storage)	231,600	57,900	28,950	7,720,000 129 NSEC	1,930,000 518 NSEC	965,000 1.036µs
2 RADIAL LINKS & 2 PERIMETER LINKS & 1 SUBSCRIBER LINE (All T1)	SINGLE - REDUNDANT (Double Storage)	463,200	115,800	57,900	4,632,000 215 NSEC	1,158,000 863 NSEC	579,000 1.727µs
	FULLY REDUNDANT (Five Times Storage)	1,158,000	289,500	144,750	1,544,000 647 NSEC	386,000 2.590µs	193,000 5.181µs
14 LINK TANDEM NODE	NON-REDUNDANT (Single Storage)	648,480 Words	162,120	81,060	21,616,000 46 NSEC	5,404,000 185 NSEC	2,702,000 370 NSEC
8 RADIAL LINKS & 9 BACKBONE LINKS (All T1)	SINGLE - REDUNDANT (Double Storage)	1,129,960	324,240	162,120	10,808,000 92 NSEC	2,702,000 370 NSEC	1,351,000 740 NSEC
	MULTI-REDUNDANT (Five Times Storage)	3,242,400	810,600	405,300	4,632,000 215 NSEC	1,158,000 863 NSEC	597,000 1,727µs
	FULLY - REDUNDANT (14 Times Storage)	9,078,720	2,269,680	1,134,840	1,544,000 647 NSEC	386,000 2.590µs	193,000 5.181µs

*Assumes triple input buffering to accommodate 1 frame delay.

5120-76E

Reference to Table 7-1 shows that a 14 link node would have to handle 21,616,000 bits per second for a input/output rate of 43,232,000 bits/second. If these bits were to be input and output serially, a memory cycle time of 23 nsec. would be required. This is beyond the current state of the art. The fact that the data will be (triple) buffered eliminates accessing conflicts between input and output, thus cutting the timing requirement in half, to 45 nsec. Grouping of bits into words for parallel I/O also proportionately eases the timing requirements. For example, use of a 4-bit word length results in a required memory cycle of 4 x 46 187 nsec.

Another method of easing timing requirements that has been considered is redundant input storage. Carried to the extreme, input data could be stored simultaneously in separate memories, one for each link, thus reducing requirements such that each memory need handle only the bit rate on one link, 1,544,000 bits/sec. This equates to a 647 nsec. cycle time for the bit serial case and 2.590 usec for the 4-bit parallel case. For the maximum 14-link configuration with an 8-bit word length and non-redundant storage, it can be seen that a data memory cycle of 304 nsec. is required. This result is entirely consistent with current state-of-the-art solid-state hardware memories having a cycle time of 215 nsec. and is fully described in Section 8 of this report.

7.3.4 Operating Systems and Systems Maintenance

7.3.4.1 Operating Systems

Each of the processing elements have been presented in the preceding paragraphs and various details of the operating systems have been discussed.

The basic architecture of each of the processors is proposed to be a background/foreground structure which utilizes background time for non-real time processing functions such as statistics, maintenance and list processing. As described, the link input and output processors background mode processing is interrupted to "service" their respective real time sequencers and to receive internal messages over the link and nodal unidirectional registers. All output microprocessors are also interrupted to process their incoming data bus FIFO stack. Nodal processors may be interrupted by all port processors to receive maintenance, statistics and traffic data as it pertains to link activity.

Initialization procedures for all links may be simplified by a master initialization approach whereby the nodal processor automatically initializes each link by transmittal of appropriate program constants that may be pre-administered. Such constants could include starting addresses for Class I frame storage, SOF synchronization patterns and initial set up of Class II list pointer tables.

7.3.4.2 System Maintenance

System maintenance falls into the two general categories of node and link oriented functions. Link maintenance, apart from any internal diagnostic functions, would provide for:

a. Monitoring synchronization
b. Timed communication tests
c. Service removal/restoration.

An out-of-synch condition may be determined by examination of the link synchronizer flag (F1) or by the timed absence of validly received packets. The latter method may provide more rapid detection due to the latitude of the synch correlation technique. Should the times absence of packets exceed a threshold value, an idle pattern can be returned for the Class I region and the link synchronizer switched to the "Request for Synch" pattern (SOF2). The input connect map and output linked list would not be modified for this condition.

If transmission fails in one direction of a link, validly received packets on that link will not be acknowledged since this would require quasi-associating such messages via another link and would lead to special packet format considerations and raise the possibility of acknowledging acknowledgements.

Timed communication tests provide validation of proper link service and these tests may be initiated by the nodal processor. Failure to pass such tests would result in an out-of-service condition, removal from service and notification of status to SYSCON or a local attendant position.

Automatic restoration of service could be provided by transmission of periodic "keep-alive" packets whereby valid reception of a pre-determined number of packets would result in link initialization.

Nodal maintenance, in addition to supporting the above, provides diagnostic fault detection, status reporting and automatic switchover to redundant processing elements. Inclusion of a maintenance panel could facilitate the service removal/ restoration of each link by system switch sensing.

As a diagnostic aid, a small residual debugging aid program is proposed as part of the final operational program for the nodal processor. Such a program would then permit on-line examination and change of system parameters for more rapid fault detection and performance monitoring.

SECTION 8

SYSTEM PROCESSOR ARCHITECTURE

SECTION 8

SYSTEM PROCESSOR ARCHITECTURE

8.1 PROBLEM

The challenge confronting the processor architecture for the SENET-DAX concept is based on both the required system dynamics and the high data rate. Through the use of state-of-the-art devices and techniques, this challenge can be effectively met.

8.2 OBJECTIVES

A high node transparency to data coupled with nodal reliability, flexibility and expandibility are the desired goals. To help in achieving these goals, data transfer should be non-redundant and non-blocking. The extensive use of subsystem modularity and parallel processing would also assist in meeting these goals.

8.3 ANALYSIS AND RESULTS

8.3.1 Microprocessors: The State-of-the-Art

8.3.1.1 Size

The state-of-the-art in microprocessor technology is experiencing continual change. Bipolar technology has produced slice-oriented chip families based on 2-and 4-bit slices, stackable to achieve desired modular widths. These modular families usually include an arithmetic logic unit (ALU), I/O interface, microprogram, and memory chips.

MOS technology now offers PMOS, NMOS and CMOS microprocessor chip families of 4-, 8-, 12- and 16-bit capabilities. These families each contain a central processing unit (CPU), I/O interface, and memory chips. Recently I^2L technology has expanded this category.

The newest dramatic advance in microprocessors is two-fold. An emitter coupled logic (ECL) slice oriented family is being introduced to allow very high speed processing with micro size. Secondly, modified NMOS 8-bit microprocessors will soon be available which contain a CPU, clock, I/O interface, and ROM/RAM memory all on the same chip. For this chip, a power supply is its only external requirement for operation.

8.3.1.2 Speed and Interrupt Capabilities

The operating speed of microprocessors today covers several orders of magnitude. In the area of MOS devices, instruction cycle times range from 0.3 - 34 usec with an average of about 3 usec. I^2L offers instruction cycle times in the order of 1 usec while bipolar microcontrollers and low power Schottky devices decrease these times to less than 500 nsec. A bipolar ECL slice oriented family will soon allow a speed of 55 nsec.

A parallel consideration in many control situations is interrupt capability. Many microprocessors allow for both software and hardware interrupts with the latter being the most important. The hardware interrupt may cause the processor to scan a specific set of registers under software control or gate logic may point to a specific register which requires processing. The second approach offers a more powerful and real time reaction and consequently has been partially incorporated into one microprocessor for 16 priority encoded interrupt levels.

8.3.1.3 Flexibility and Processing Power

Processing power and control flexibility are frequently inversely related for any one device. A micro-controller provides a very limited set of 8 instructions for powerful and fast bit manipulation in specific applications. MOS microprocessors offer sets of 28 to 156 instructions with an average of 75. These devices have an average instruction cycle time of only 3 usec but can effectively service a very wide range of applications.

When considering an I^2L device, the disparaties between processing power and control flexibility lessen while bipolar low power Schottky families optimize both parameters. These bit slice families couple high-speed processing with an average set of 40 instructions to provide a good control flexibility.

8.3.1.4 Reliability

The high reliability of LSI devices has been one basic reason for their continued production. The minimization of discrete connections always increases system reliability.

In a computer system where many operations are performed sequentially by one central processor (CPU), a complete back-up CPU must operate in parallel to assume control if the on-line CPU fails. However, if each operation or group of operations are performed in parallel by a less powerful but separate microprocessor, each operation could be performed more slowly yet the total time to completion could be significantly less than when sequential operations are performed. Additionally should a single microprocessor fail most other operations could still function.

The assertion could be made that parallel processing with dispersed processors creates more discrete connections and thereby lowers system reliability. However, central processor systems need large complicated memory and bus structures for total program and temporary data storage. In comparison, dispersed processors with associated memory can be physically separate for each function.

8.3.1.5 Memory Capabilities

Direct memory addressing capability is an important consideration for microprocessors. Bit slice chip families allow open ended memory expansion dependent mainly on the users own microprogram and additional bit slice hardware.

Most 8- and 16-bit microprocessors can directly address 65K bytes. Many addressing modes are available in varied combinations including the following: direct, extended, indexed, relative, inherent, and content addressable. An indication of the processing power of a system control element is obtained through knowledge of its memory capabilities.

8.3.1.6 Power Consumption

Technological advances have generally reduced semiconductor speed/power products, but power consumption for a system continues to be directly proportional to its processing capabilities and speed. Monolithic microprocessor design has reduced the power requirements inherent in large computers where power drivers are needed to transfer data between system elements over high capacitance buses.

In a system of dispersed microprocessors, many CPU's operate simultaneously although the speed/power product for each device is far below that of one central processor. Low power Schottky designs have reduced bipolar speed/power products significantly with recent contributions in this area being made by I^2L.

8.3.1.7 Summary

The microprocessor field is a very diversified and rapidly expanding area of semiconductor technology. In application, these devices are currently utilized in multiples by computer companies to construct large high power machines. As a result of dramatic cost improvements, microprocessor cost has been reduced to the level where they are now available even to the home hobbyist.

The multiplicity of mocroprocessors available has created a dilemma for the engineer. To assist in effectively selecting a microprocessor for a given application, a priority consideration of system parameters should be established. These parameters should include the following areas: interrupt types, memory size and modularity, space, power, bit manipulation, cost and, data processing and byte transfer speed requirements. Once provided with a priority structured listing of these parameters, the engineer's dilemma is considerably reduced.

The varied SENET-DAX concept requirements of speed, flexibility, reliability and expandibility appear to be optimally satisfied in the utilization of a distributed microprocessor/microcontroller system. To assist in achieving these goals, modularity will be utilized at all control levels.

8.3.2 Nodal Architecture

The SENET-DAX nodal function is the efficient inter-port switching of both high-speed variable length subscriber/data packets and multi-rate local voice/data subscribers. As seen in Figure 8-1 the focal point of nodal activity is the redundant tri-level common bus. Pathways to the common bus are provided for redundant nodal control elements and a maximum of 16 trunk links or subscriber access ports. A bus port will service one trunk link or approximately 300 local subscribers (see Section 11.1).

8.3.2.1 Decentralized Control

The SENET-DAX system has been examined for a maximum nodal service capability of 14 T1 trunk links and approximately 600 local subscribers (Figure 8-2). In many situations the nodal switch will only be required to service several trunk links. If one large high power computer was utilized for all nodal switching functions, inefficiencies would be evident in unused processing time (power consumption constant) and unnecessary hardware overhead.

These problems effectively vanish when the majority of link and access control functions are placed at the link/access level. In a distributed control structure the nodal processor performs routing table maintenance, diagnostic routines and attendant console servicing. This structure isolates the nodal processor from real time data operations and increases nodal reliability through modular control. The minimized nodal control functions are few enough to allow the application of a microprocessor for nodal control.

8.3.2.2 Decentralized Memory

Modular expansion of memory is desireable in a flexible and dynamic system. Efficient memory utilization requires non-redundant data storage and dynamic allocation. A maximum of 16 near simultaneous high-speed read/write operations on a common memory represents severe memory modularity.

A port memory module appends every port controller. This memory is adequate to store average incoming busy hour traffic to that port. The physical attachment of a control/memory module to the nodal bus at a port specifies its memory element address in augmenting the total nodal memory. If additional memory is required by a port controller, the nodal processor may assign a maximum of 8 port controllers to temporarily utilize dynamic portions of the nodal overflow memory (Figure 8-2).

The port memory is dynamically allocable by the port controller. Following the storage of incoming data in memory, the port controller obtains internal routing directions from the routing table to determine which port is to transmit that data. The transmitting port controller is given the address of the stored data for real time retrieval and incorporation into its transmitted serial bit stream.

8.3.2.3 Nodal Timing

Non-blocking simultaneous data transfers require strict synchronization. Figure 8-3 depicts nodal timing based on parallel 8-bit byte data transfers.

The 8-bit byte sizing resulted from many considerations. A T1 data frame length of 10 msec was established: to minimize memory requirements for three frame Class I storage; to minimize the total contiguous signaling message load to an input processor; and to allow rapid link signaling responses. The 10-msec frame length and

8-bit data transfer only require subscriber interface bit stuffing for one of many anticipated subscriber bit rates. The byte also corresponds to the smallest bit grouping for trunk link signaling flags.

Three timing phases are contained in each bit time so 8-bit times provide 24 discrete time positions. The selection of three phases per bit time was dictated by a solid-state memory having a cycle time of 215 nsec. Generally, phase A allows input data real time byte transfers into individual port memories, while the remaining 16 phases (B and C) sequence all port controllers in reading total node memory. Table 8-1 indicates timing control of the tri-level common bus.

Redundant nodal timing is intentionally omitted in Figure 8-2. Depending on the level of reliability desired, the timing could be made redundant either in the nodal control or per port with a nodal sync supplied. The latter would assist in minimizing port controller-to-node connections.

Assuming a possible modular port memory size of 8K bytes and a maximum Class II packet length of 256 bytes, Table 8-2 indicates the primary signals on each level of the tri-level redundant bus structure. The number of redundant address and data lines required per bus level is also specified.

8.3.2.4 Single-Thread Elements

Single-thread elements are node components whose reliability affects nodal performance. These nodal hardware elements are redundant (Figures 8-1 and 8-2). Future analysis may allow reductions in redundant nodal hardware.

8.3.2.5 Expandability and Flexibility

Concepts pertinent to nodal expandability and flexibility are contained in Section 8.3.2. The decentralized control/memory per port provides total nodal versatility. A totally tandem node is feasible, upper bounded by the 16 available ports. A node servicing a single link and a single subscriber access port is conceivable, and expandable to 16 for either access or link purposes.

8.3.3 Link Processing Architecture

Efficiency in link control requires input and output processor isolation from real time data manipulation.

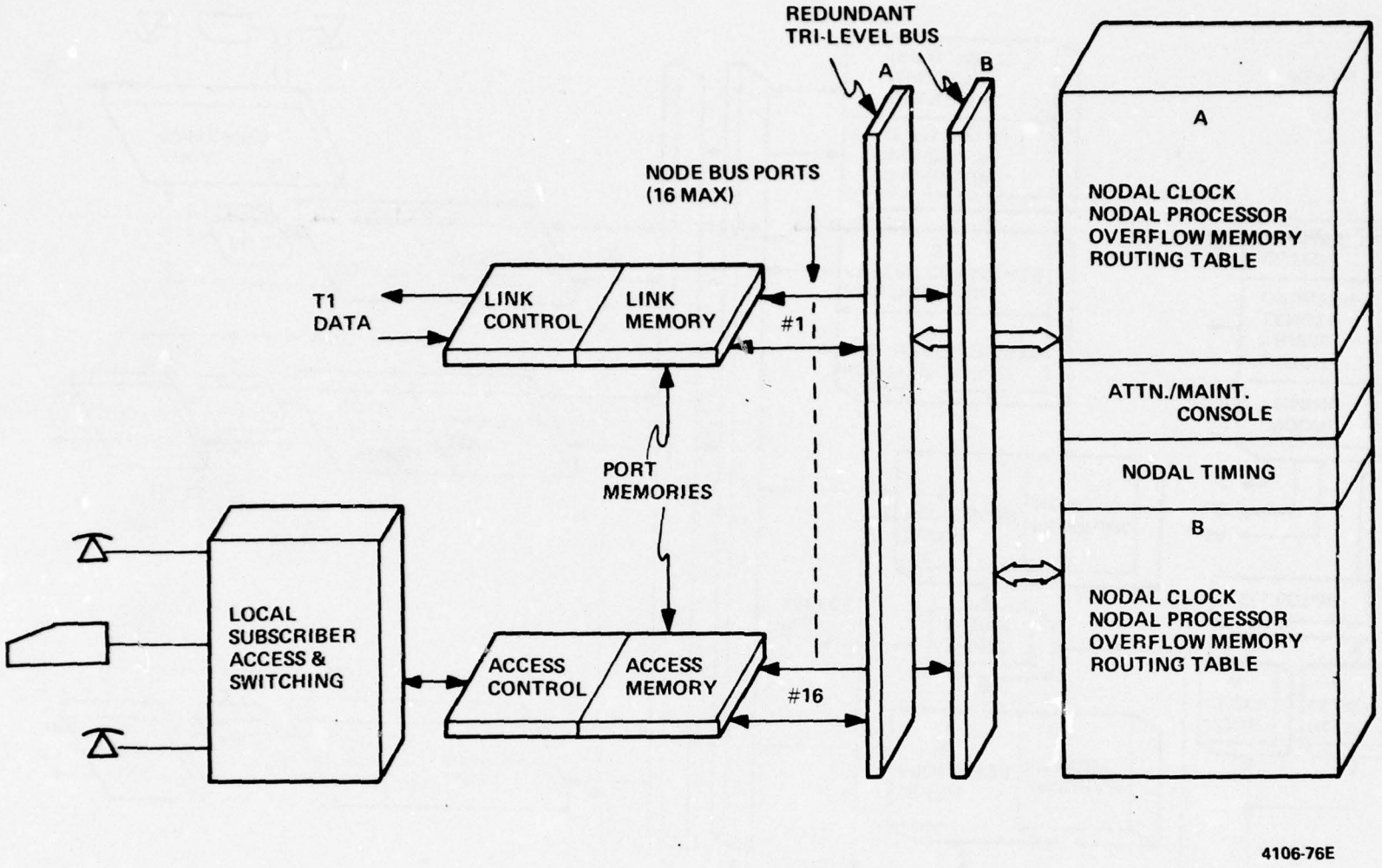


Figure 8-1. SENET-DAX Node

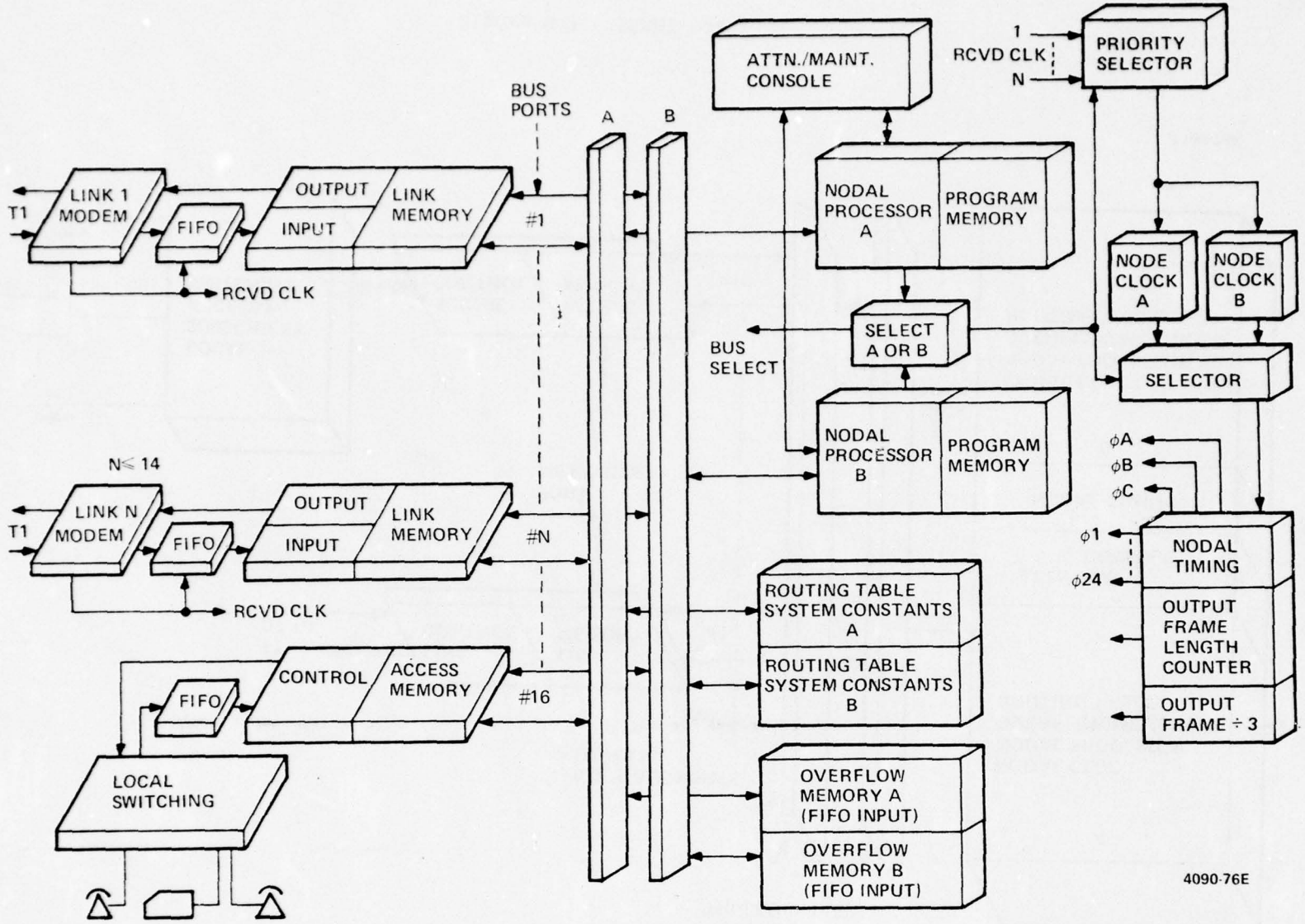


Figure 8-2. Node Processor Architecture

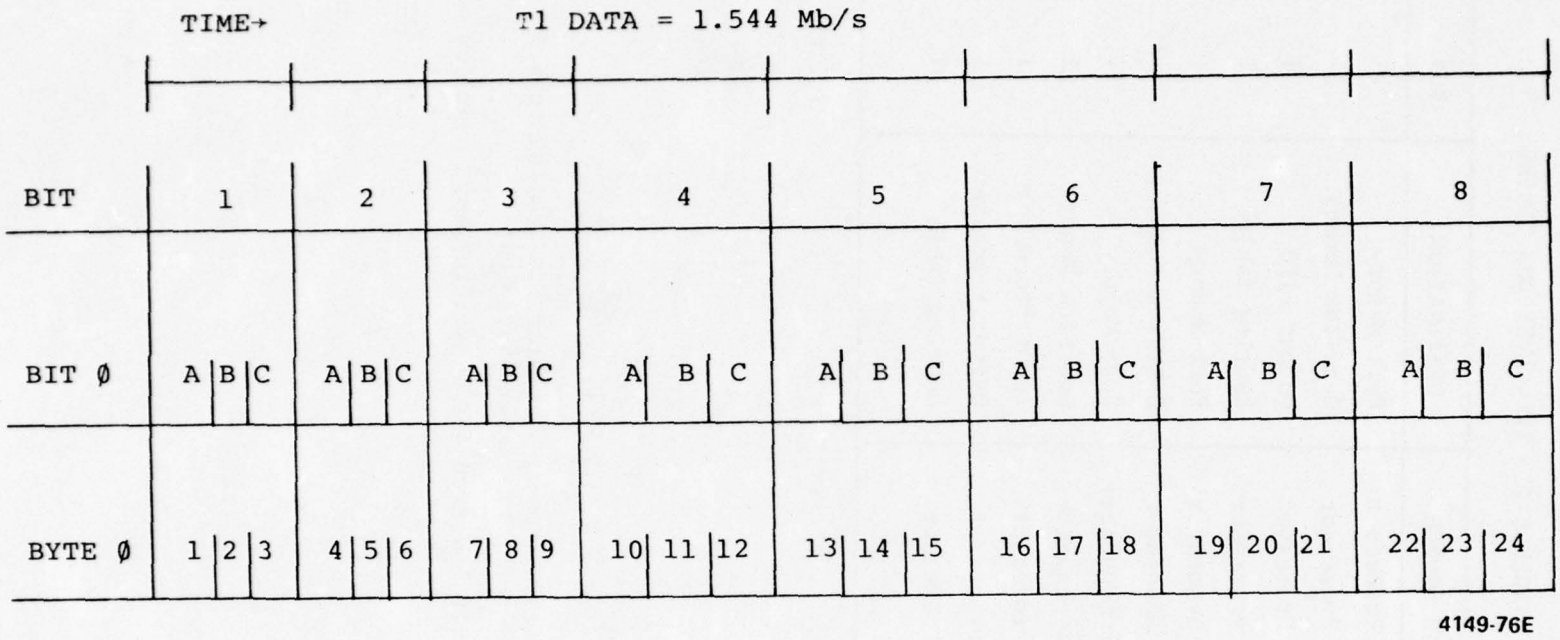


Figure 8-3. Nodal Timing

TABLE 8-1. TRI-LEVEL BUS TIMING

ITEM	COMMAND SOURCE	DESTINATION	BUS	0	
1	Input Processor	Port Memory	-	A + t	(1)
2	Input Processor	Overflow Memory	II	A	(2)
3	Input Processor	Output FIFO	I	B or C	
4	Input Processor	Routing Table	I	(B or C) +8t	(1)
5	Input Sequencer	Port Memory	-	A	
6	Input Sequencer	Overflow Memory	III	B or C	(3)
7	Output Sequencer	Port Memory	II	B or C	
8	Output Sequencer	Overflow Memory	II	B or C	
9	Node Processor	Input Processor	I	A	
10	Node Processor	Output Processor	I	A	
11	Node Processor	Routing Table	I	A	

1. $t = (1.544\ \text{MHz})^{-1}$
2. Overflow memory utilization limited to 8 ports simultaneously by nodal processor assignment
3. FIFO buffered for ØA writing into overflow memory.

TABLE 8-2. *TRI-LEVEL BUS SIGNALS AND LINES*

BUS LEVEL	PRIMARY SIGNALS	ADDRESS LINES	DATA LINES	SUB-TOTAL
I	Nodal Processor		3 Precedence	
	Control, Internal		13 + 6 Address	
	Class II Routing		8 Byte Count	
		6	30	36
II	Output Sequencer Reading Nodal Memory	13 + 6 = 19	8	27
III	Overflow Memory Write	13	8	21

Total Lines In Each Redundant Tri-Level Bus = 84

8.3.3.1 Microprocessors for Parallel Processing

Byte transfers of accumulated input T1 data bits must occur every 5 usec. A real time processor would have difficulty performing this function along with port memory allocation, free list generation, signaling analysis, Class I field length updates, and statistical and diagnostic reports to the nodal microprocessor. Consequently the real time function is performed by a sequencer slaved to the input microprocessor. This allows input microprocessor bit sizing to be dictated by other processing considerations. An important input microprocessor function is internal CCIS message routing. After determining which output microprocessor is to transmit a CCIS message, the input microprocessor loads a FIFO attached to the output microprocessor with the message precedence, nodal memory location and, message byte count (see Figure 8-4 and Table 8-1, Item 3). With few bus and FIFO width considerations, a flexible system precedence structure is software programmable.

Output processing real time restraints also require a real time output sequencer slaved to the output microprocessor. The output microprocessor must perform the following functions: Class I link connection map management, Class II packet management for transmission by precedence and, timeout monitoring of transmitted Class II packets for possible retransmission.

Link microprocessors communicate together through unidirectional buffers with nodal timing control. Communication between the nodal microprocessor and the two link microprocessors is also by unidirectional buffers.

8.3.3.2 Real Time Data Transfer Control

Bipolar bit slice microprocessor families utilize a microprogrammer/sequencer chip. This element is used twice in the SENET-DAX link controller to either write or read under microprocessor control.

The input sequencer collects a data byte and stores it at a port memory address. The port memory address is taken from one of three free lists provided by the input microprocessor for Class I, Class II CCIS and data packets. The link synchronizer directs the input sequencer to each list and correlates Class II packet frame check sequences (FCS). If a packet delimiter flag and frame check sequence concur, the input microprocessor is informed of the valid packet first byte address and number of packet bytes.

The output sequencer utilizes an address list formed by its output microprocessor. The list successively contains nodal memory addresses accompanied by a packet length byte count. As a result of memory timing the sequencer may read the nodal memory only once during an 8-bit period. Therefore, concurrent with the link synchronizer transmitting a start-of-frame (SOF) or delimeter flag (Class I, Class II, respectively) the output sequencer extracts from nodal memory the first byte of the first Class I subscriber or Class II CCIS/data packet for transmission. While that byte is serially transmitted the sequencer has 8-bit periods to extract the second byte. This sequence continues for n byte counts. For Class II packets the link synchronizer then transmits the frame check sequence it produced during packet transmission and appends the FCS with the delimiter flag.

Class II data bit stuffing ("0" bit insertion in the serial data stream following five contiguous "1's") is performed by hardware on all Class II bits except delimiter flags. Correspondingly for input data, hardware "destuffs" Class II prior to byte storage.

8.3.3.3 Link Sync and Error Control

The link synchronizer provides synchronization and error control for all link data. Start-of-frame (SOF) flags are extracted from incoming data for successive correlation (see Section 9.2). An SOF flag is transmitted every frame, type based on the currently correlated input sync condition.

Class II delimiter flags are both detected and transmitted appropriately. Similarly, frame check sequences (FCS) are correlated and produced.

Status monitoring of the input data FIFO buffer is performed by the link synchronizer. Following initialization to half full the buffer should, on the average, maintain that condition. If the node switch is forced to rely on its quartz oscillator clock over an extended time period the buffer may overflow or underflow. Buffer status warns of this impending condition, allowing the link synchronizer to reinitialize the buffer.

The detection and production of Class II delimiter flags and FCS are hardware functions. SOF correlation and production along with FIFO buffer status monitoring are software functions. These software functions might merit either a separate microprocessor or incorporation into the link output microprocessor program.

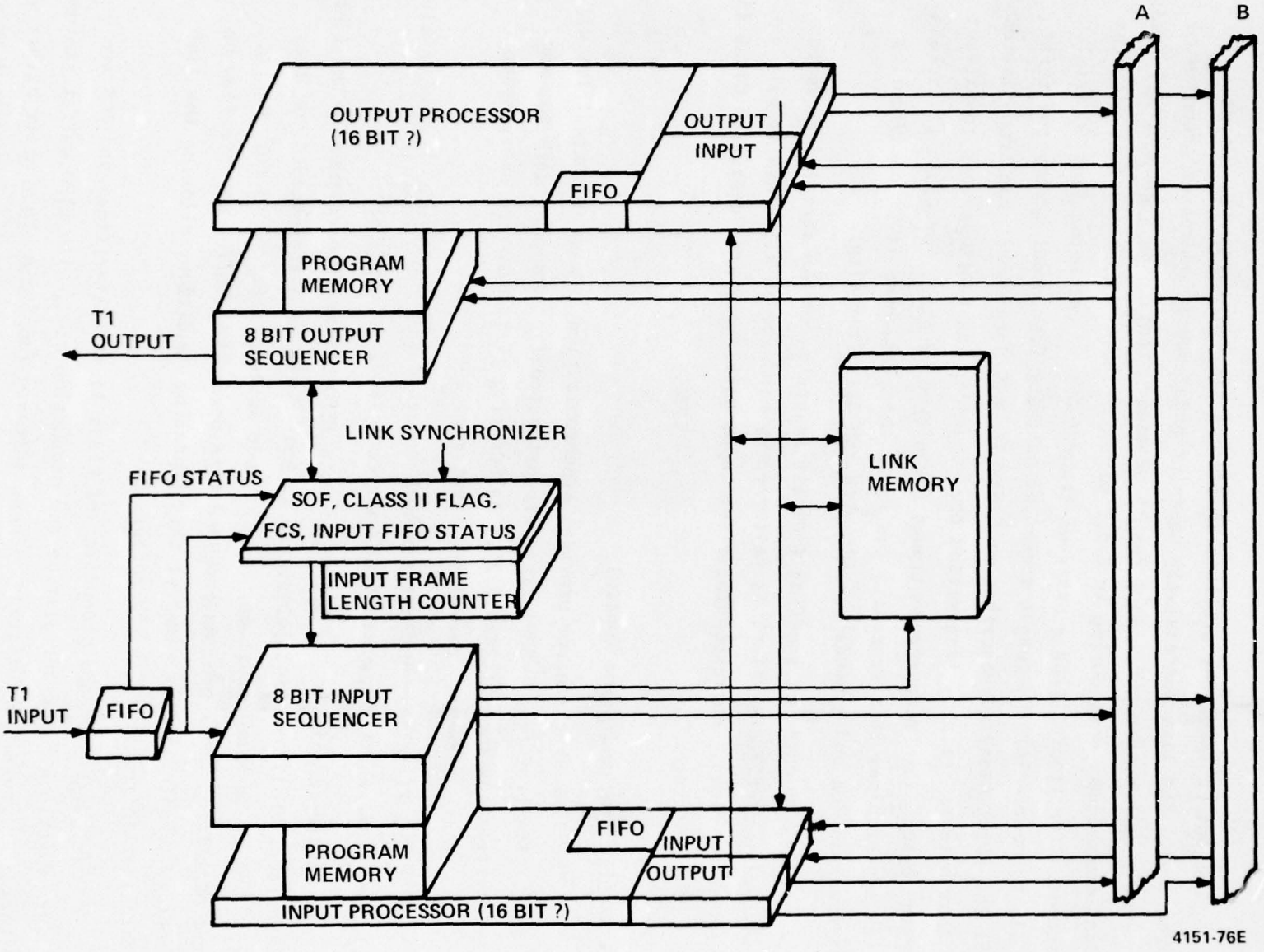


Figure 8-4. Link Processing

8.3.4 Subscriber Access Architecture

Microprocessor control for an access port appears to be relatively simple. The access port control element indicated in Figure 8-2 might also control local subscriber switching. Control element duties would include access memory management, CCIS packet management, and transfer of data bytes from nodal memory to subscriber interface output registers.

A possible architecture for subscriber access is shown in Figure 8-5. Coincident with subscriber call allocation, the access port microprocessor provides the subscriber interface input register with a starting address and byte length of access port memory reserved for that call. The interface register collects subscriber bits and parallel transfers bytes and incremented byte addresses through a FIFO buffer directly into access port memory.

The use of microprocessors and register/ROM's for each subscriber termination is attractive from the standpoint of the evolving low cost of microprocessors and associated circuitry. However, it is only one possible implementation. Architectures utilizing one microprocessor for several terminations, or digital logic, or firmware, or some combination of these, remain a future area of investigation.

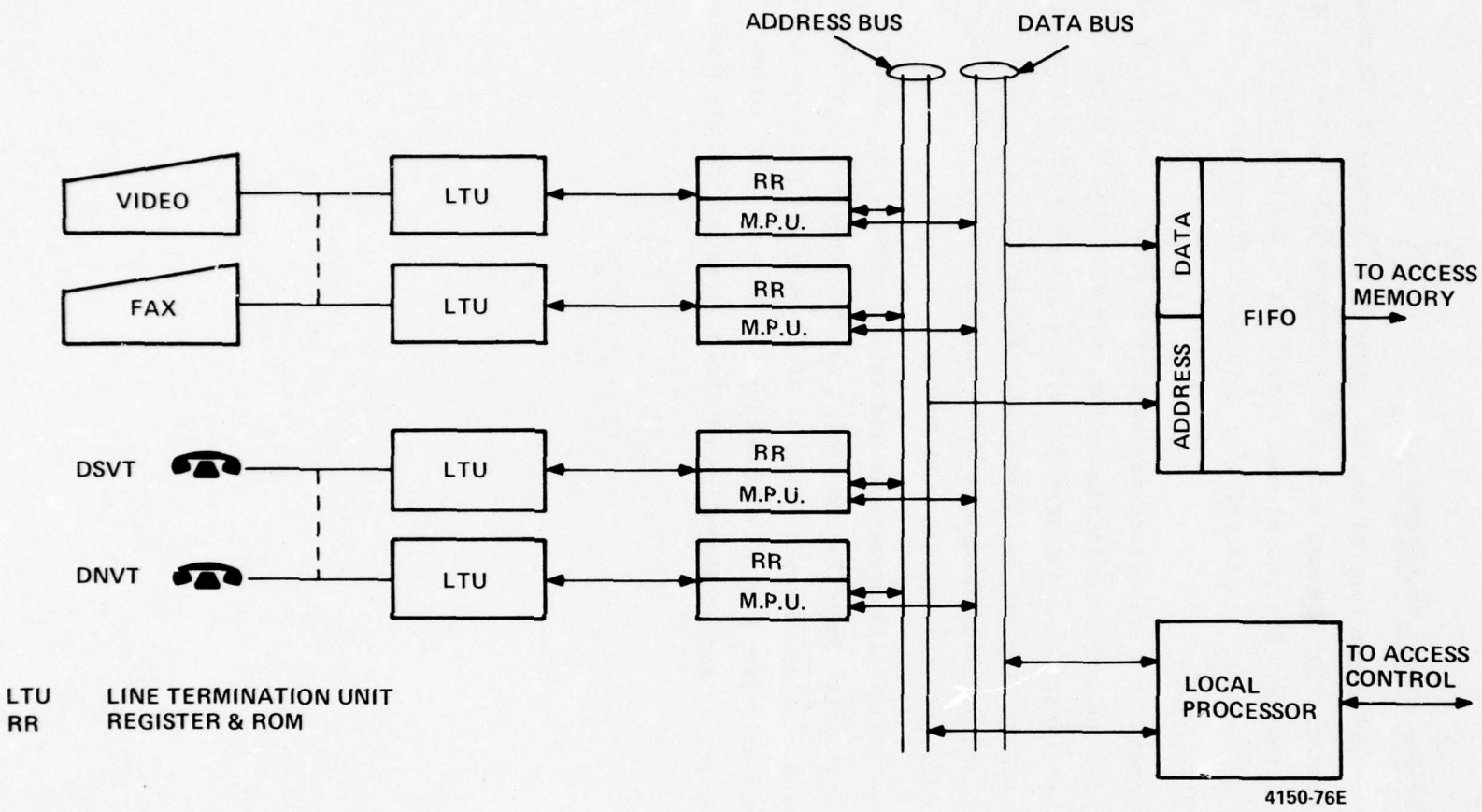


Figure 8-5. Subscriber Access Microprocessor Approach

SECTION 9

SYNCHRONIZATION TECHNIQUES

SECTION 9

SYNCHRONIZATION TECHNIQUES

9.1 BIT SYNCHRONIZATION

9.1.1 Problem

In Section 4.4 the major network synchronization techniques are examined with respect to their ability to provide effective communications between DAX's. It is shown there that with the single exception of bit stuffing, any of the synchronization techniques considered would be applicable to the DAX network. In addition, it is shown that the choice of an "optimum" network synchronization plan is directly dependent on overall network performance objectives since the performance level of each synchronization plan is functionally related to the following network evaluation factors: cost, reliability, survivability, and complexity. It follows therefore that the choice of an "optimum" synchronization technique will change as the weighting assigned to each evaluation factor changes.

With this as background, the problem at hand consists of the following:

a. To choose a synchronization plan which best satisfies the objectives set forth in the next section
b. To develop a conceptual design which realizes the functional and technical requirements of the chosen synchronization plan
c. To evaluate any hardware/software tradeoffs inherent in the conceptual design.

9.1.2 Objectives

The network synchronization plan chosen must reflect the network evaluation factors listed above. The following listing of weighting factors, in decreasing order of importance, is offered:

a. Survivability
b. Cost
c. Reliability
d. Complexity.

In addition to these four evaluation factors, there exist other pertinent factors which must be considered when choosing a synchronization plan. These additional factors derive from both the nature of the DAX concept and the environment in which it is likely to be employed. They are:

e. Transparency - The software requirements of a DAX are extensive due primarily to the dynamic nature of the master frame and to a lesser extent, the subscriber features provided. Therefore, as a general design philosophy, minimization of software within the goals of the system is to be considered a prime objective. To this end, interaction between a DAX timing subsystem (i.e., the portion of the network synchronization system located at each DAX.) and its controlling processor (whether it be the main CPU or a time shared microprocessor) or between DAX's (e.g., exchanging control information for synchronization purposes) is to be minimized. In effect, the network synchronization system should operate independently of other network functions to the maximum extent possible, and all DAX synchronization subsystems should operate independently of each other.

f. Network Compatibility - The DAX will likely be employed in a network possessing a hierarchical structure. For the purposes of this study the assumed network configuration is the singly spoked wheel structure described in the Appendix. This network is a 2-level hierarchy with the higher level tandem nodes being fully interconnected. Such an arrangement lends itself to certain synchronization techniques which can be incorporated into the conceptual design.

9.1.3 Analysis and Results

9.1.3.1 Synchronization Technique Selection

A consideration of the network structure reveals that the modified master/slave synchronization technique is well suited to the postulated DAX network.

Because of this and the overall ability of the master/slave technique to satisfy network performance objectives, it is the technique that will be considered further in the conceptual design. In this approach, each tandem node in the wheel configuration would serve as a local master for all regional nodes (slaves) to which it is directly connected; thus, each local master and its connected slaves would form, in effect, an independent subnetwork. For the particular wheel configuration illustrated in the Appendix (i.e., 10 tandem nodes and 50 regional nodes), the DAX network would comprise 10 subnetworks with each subnetwork containing 5 slave nodes. This is illustrated in Figure 9-1. Observe that all subnetworks are interconnected at both the local master and slave levels. It is assumed that each subnetwork would be defined by geographical as well as traffic considerations.

As just described, within each subnetwork all regional nodes would slave their timing to the tandem node. Thus each subnetwork would operate synchronously and essentially slip-free. This type of operation provides a distinct advantage over independent clocks since it does not require that link buffers be periodically reset. Although the modified master/slave does not provide the survivability derived from independent clocks, it still affords a reasonable degree of survivability because of assumed extensive connectivity in a DAX network.

There are three alternatives for synchronizing the local masters: frequency averaging, a master/slave arrangement, or independent atomic clocks. The first two approaches would result in an overall synchronous network. That is, all nodal clocks (master and slave) would maintain the same long-term average frequency. The independent clock approach would result in asynchronous operation between subnetworks and as previously mentioned, synchronous operation within a subnetwork. Again, consideration of the assumed network structure, specifically the fact that the tandem nodes are fully interconnected, reduces the selection to frequency averaging and the master/slave approach. This follows, since for the given network configuration independent atomic clocks would not be significantly more survivable than either of the other two approaches, and would definitely be higher priced. Note that even if the tandem nodes are not fully interconnected in an actual implementation, the degree of connectivity is likely to be sufficiently high to make the elimination of the independent clock approach justifiable. An additional consideration is the fact that future communications will likely see

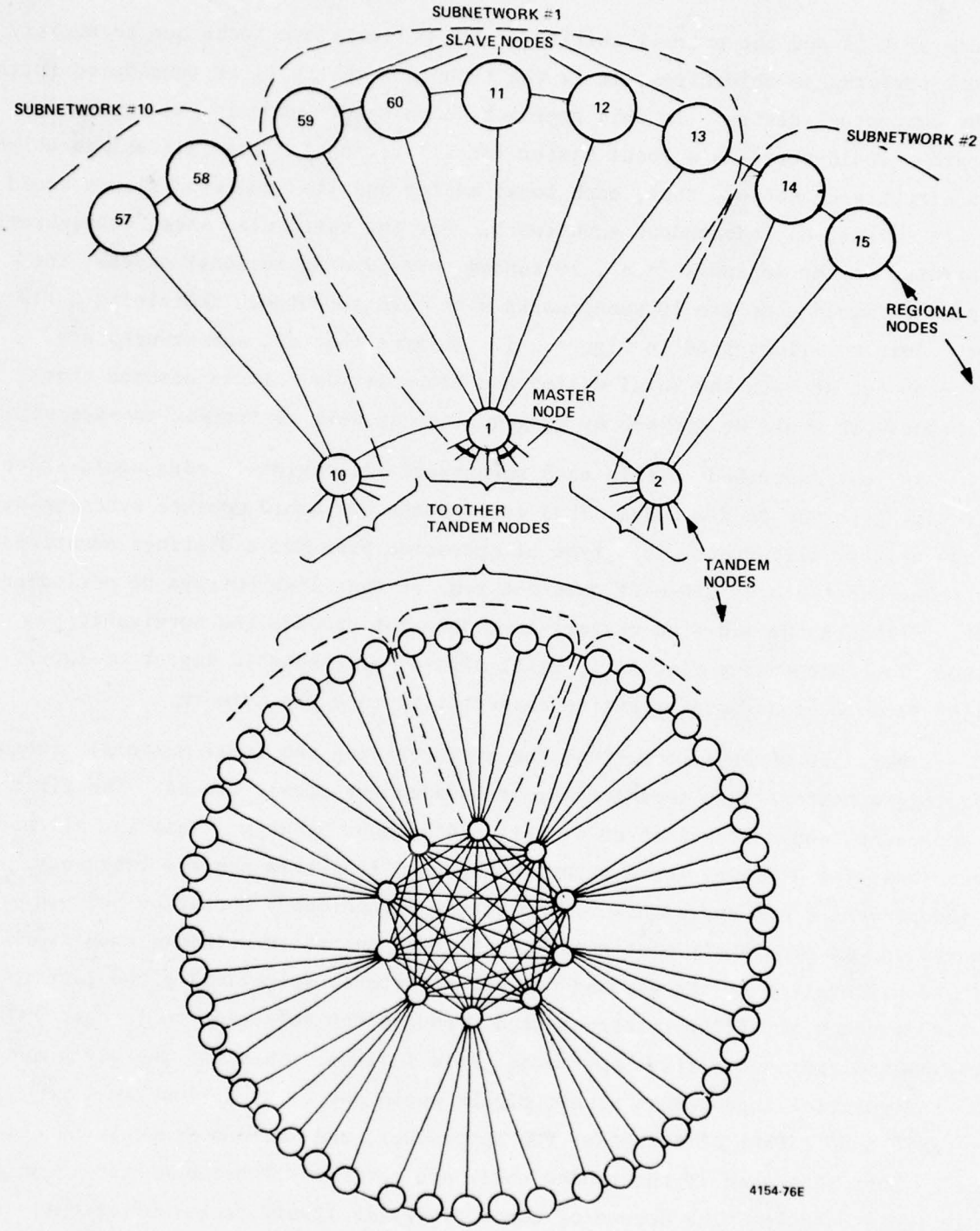


Figure 9-1. DAX Network Structure

a greater and greater use of end-to-end encryption, making asynchronous techniques less and less attractive.

The choice between the master/slave approach and frequency averaging is not so clear. It would appear, however, that the advanced state-of-the-art in master/slave techniques and the numerous stability problems inherent in frequency averaging makes the former the practical approach. For this reason, the master/slave technique is the approach that will be used to synchronize master nodes in the conceptual design. It should be noted that numerous study efforts in the area of frequency averaging are presently under way. As more experience is gained in this area, both practical and theoretical, a reevaluation of DAX network synchronization may be in order.

9.1.3.2 Conceptual Design

9.1.3.2.1 System Description - Figure 9-2 illustrates a singly spoked wheel network redrawn to reflect, possibly, geographical or traffic requirements. Observe that there are four subnetworks as defined previously, each composed of one tandem and three regional nodes. It will also be helpful to define a sixth subnetwork which is composed of all the tandem switches. This sixth subnetwork is indicated by the interconnecting dotted line in Figure 9-2. The communication paths (network links) between nodes can be any of various transmission systems (e.g., LOS radio, coaxial cable, satellite, etc.). Network timing will be disseminated via a timing tree whose links are a subset of the network links.

In accordance with the chosen synchronization plan, a particular tandem node is designated master node for subnetwork 6. That is, during normal operation, all other tandem nodes slave their timing to this node. Because the tandem nodes are fully interconnected, any tandem node could be chosen for this purpose. It is assumed for illustrative purposes that T_5 (in Figure 9-2) is designated master. It should be realized though that T_5 serves as master not only for the other tandem nodes/local masters but also for all regional nodes (slaves) via the three level hierarchy which results from the modified master/slave implementation. Within subnetwork 6 all communications will be synchronous and slip-free. This assumes

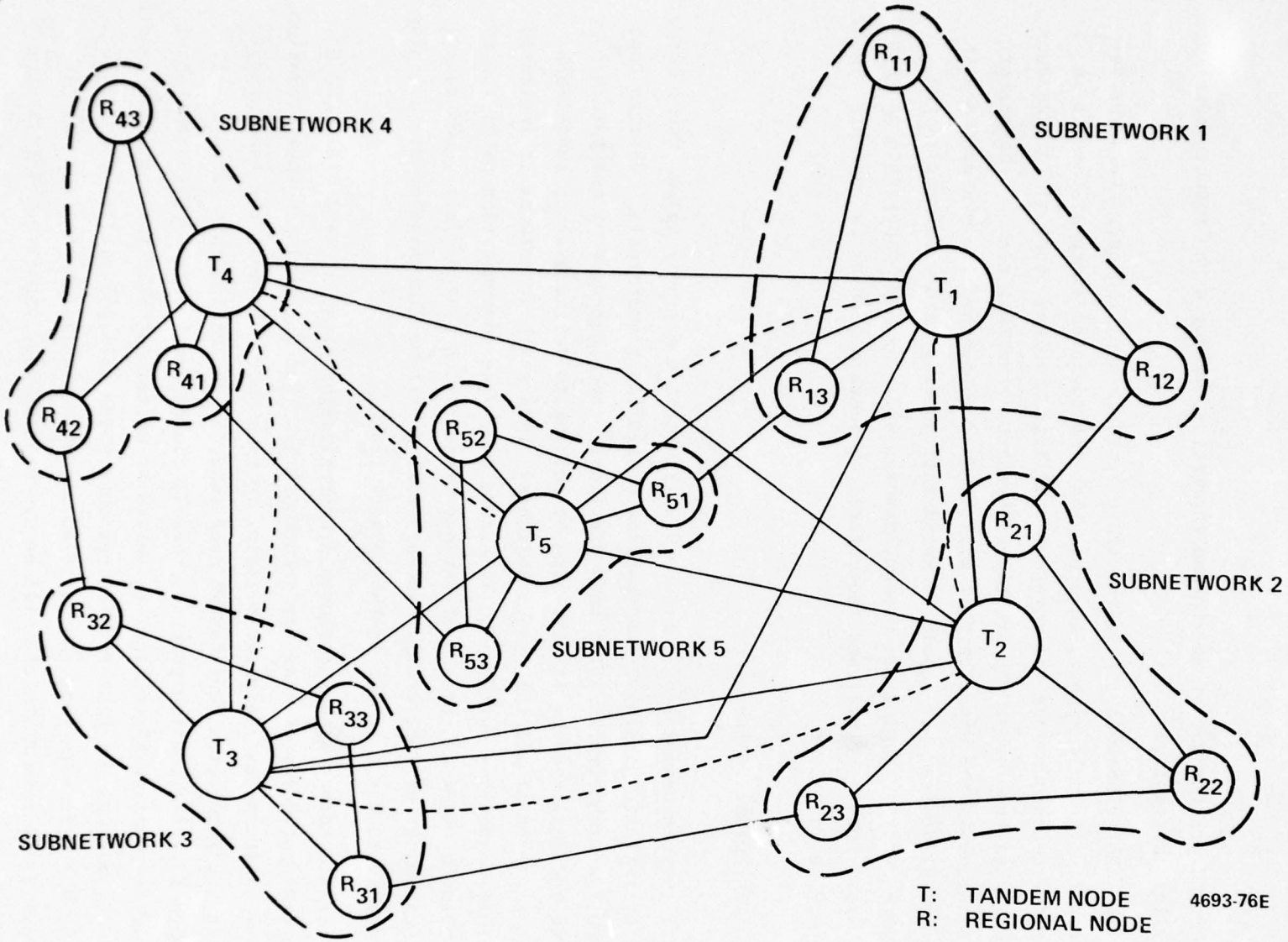


Figure 9-2. Typical Network Configuration

that each link is equipped with a buffer to absorb transmission delay variations. This subject is discussed in more detail in Section 4.4.2. Similarly, within and between subnetworks 1 through 5 communications will also be synchronous and slip-free.

As presently described, the network is extremely vulnerable to loss of the master. In fact, in order to avoid frequent link buffer overflow/underflow in this event, it would be necessary to equip all nodes with either very stable clocks, large link buffers, or an appropriate combination of the two. As a means of avoiding this necessity and to greatly improve survivability, it is proposed that at least one other tandem node be capable of assuming the master role. It would be possible to have all tandems so equipped; however, the actual number of standby masters used would be determined from the degree of survivability desired, the accuracy of the nodal clocks, and the size of the link buffers (Section 9.1.3.2.2.2 examines this tradeoff).

As a practical means of implementing this master/standby master arrangement, it will simply be necessary to assign a slaving precedence to the tandem nodes. The master would be assigned the highest precedence n (where n is one plus the number of standby masters) and the standby masters successively lower precedences (n-1, n-2, n-3, - - -, 1) as determined in an appropriate manner. Then, at any tandem node which is slaving its timing, the nodal clock would slave to the link which has the highest precedence level and from which proper timing signals are being received. Obviously, each node would require sufficient hardware/software to determine that the timing being derived from a particular source is valid. An attractive feature of this precedence approach is that the network timing tree automatically reconfigures when the master fails. For example, Figure 9-3a illustrates the timing tree during normal operation. As may be seen, T_5 is the master node and all timing is slaved directly or indirectly to this node. Assume T_1 is designated a standby master and has the second highest precedence level. Then if the nodal clock at T_5 should fail to transmit timing, the network would automatically reconfigure as shown in Figure 9-3b. Figure 9-3b illustrates a total timing tree reconfiguration. It is also possible for only part of the timing tree to reconfigure. For example, if the link carrying timing signals from T_5 to T_2 were to fail, then T_2 would automatically slave to T_1 since it is the node with the second highest precedence level. Timing in subnetworks 1, 3, 4 and 5 would be unaffected by this partial reconfiguration.

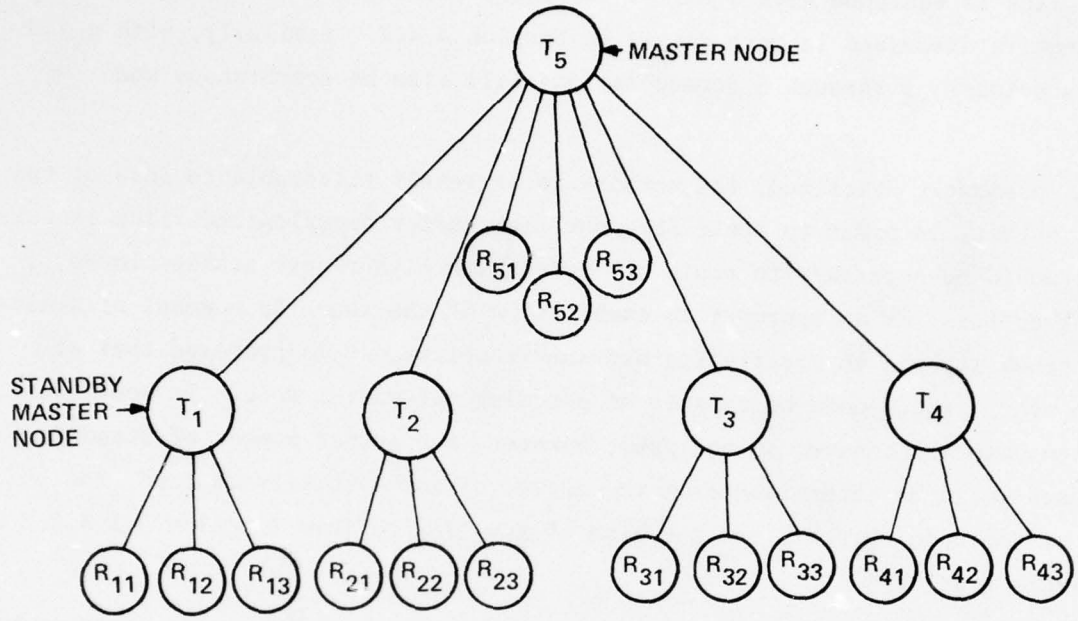


Figure 9-3a. Normal Timing Tree

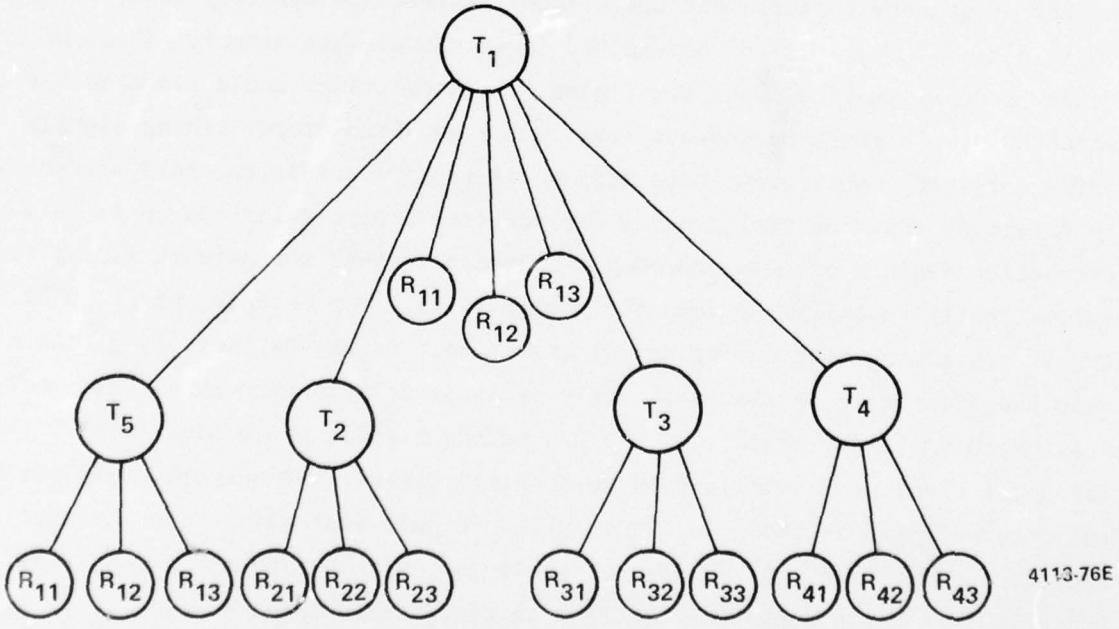


Figure 9-3b. Timing Tree After Clock Failure at T_5

The timing plan described above requires that the precedence level of all tandem nodes be stored at each tandem node. Such a requirement should present no problem. As an added network service feature, it would be a simple matter to permit altering of the assigned precedences of the standby masters via specifically designed CCIS messages. This capability would be attractive for maintenance and security reasons.

In order to provide for communications within or between subnetworks which for one reason or another are incapable of operating in a slaved node, all nodal clocks will be capable of free-running. The required stability of the nodal clocks in the free-running mode is a function of the size of the link buffers and several other parameters. This subject is explored further in Section 9.1.3.2.2. During normal operations, the link buffers at each node should maintain a long-term average fill of one-half. Short-term variations from this value are to be expected due primarily to jitter introduced by repeaters, transmission delay variations, and tracking inaccuracies in the phase locked loops. Major excursions from the half-way point would arise when a nodal clock is free-running or the phase locked loop is acquiring lock following an outage or possibly a switch from the network master to a standby master. The link buffers must be sufficiently large to absorb the maximum bit slippage to be expected during the two types of events. The more severe of the two events is the case of the free-running oscillator and the link buffers will be designed for this eventuality.

It is assumed that the master for the network uses a cesium-beam atomic clock. It is further assumed that the standby masters also use atomic clocks. However, it is not necessary for the standby masters to employ primary standards; rubidium vapor cell clocks will be adequate. If such is the case, though, it will be necessary to periodically recalibrate the rubidiums against a primary standard. This is necessary in order to ensure that a switch from the master to a standby master can be made without quickly suffering buffer overflow/underflow. The stability requirements for non-master tandem and regional nodes make high precision quartz oscillators adequate. A detailed discussion of clock stabilities is provided in Section 9.1.3.2.2.1.

9.1.3.2.2 DAX Timing Units - Figures 9-4a and 9-4b illustrate the proposed designs for the DAX timing units. The former would be employed at all slave locations and the latter at both the master and all standby master locations. It should be recalled that a slave location can be either a regional or tandem node while a master or standby master location can only be a tandem node. Both designs are essentially equivalent except for the type of nodal clocks employed. The slave configuration uses two double oven quartz oscillators and the master/standby master configuration uses an atomic clock and a double oven quartz oscillator. As discussed in the previous section, the atomic clock at the master location is a cesium-beam type while all standby masters possess rubidium vapor cells.

As designed, the DAX timing unit is self-contained and should require very little interaction with its controlling processor. However, the ability to transmit to or receive information from the controlling processor has been incorporated into the major subsystems of the timing unit. This is necessary for reliability purposes since the timing unit supplies timing for all subsystems at a DAX. The following subsections describe requirements and functions of the various timing unit subsystems.

9.1.3.2.2.1 Nodal Clocks - An ideal oscillator is characterized by an output signal of the form

$$V(t) = A \cos 2\pi f_o t$$

where amplitude and frequency are constant with time. Unfortunately, all oscillators are subject to amplitude and frequency deviations such that the actual output is given by

$$V(t) = (A + \theta(t)) \cos (2\pi f_o t + \Phi(t))$$

For high precision oscillators, $\theta(t)$ and $\Phi(t)$ are random processes with the properties that

$$| \theta(t)/A | \leq 1$$

and

$$| \Phi(t)/(2\pi f_o) | \leq 1$$

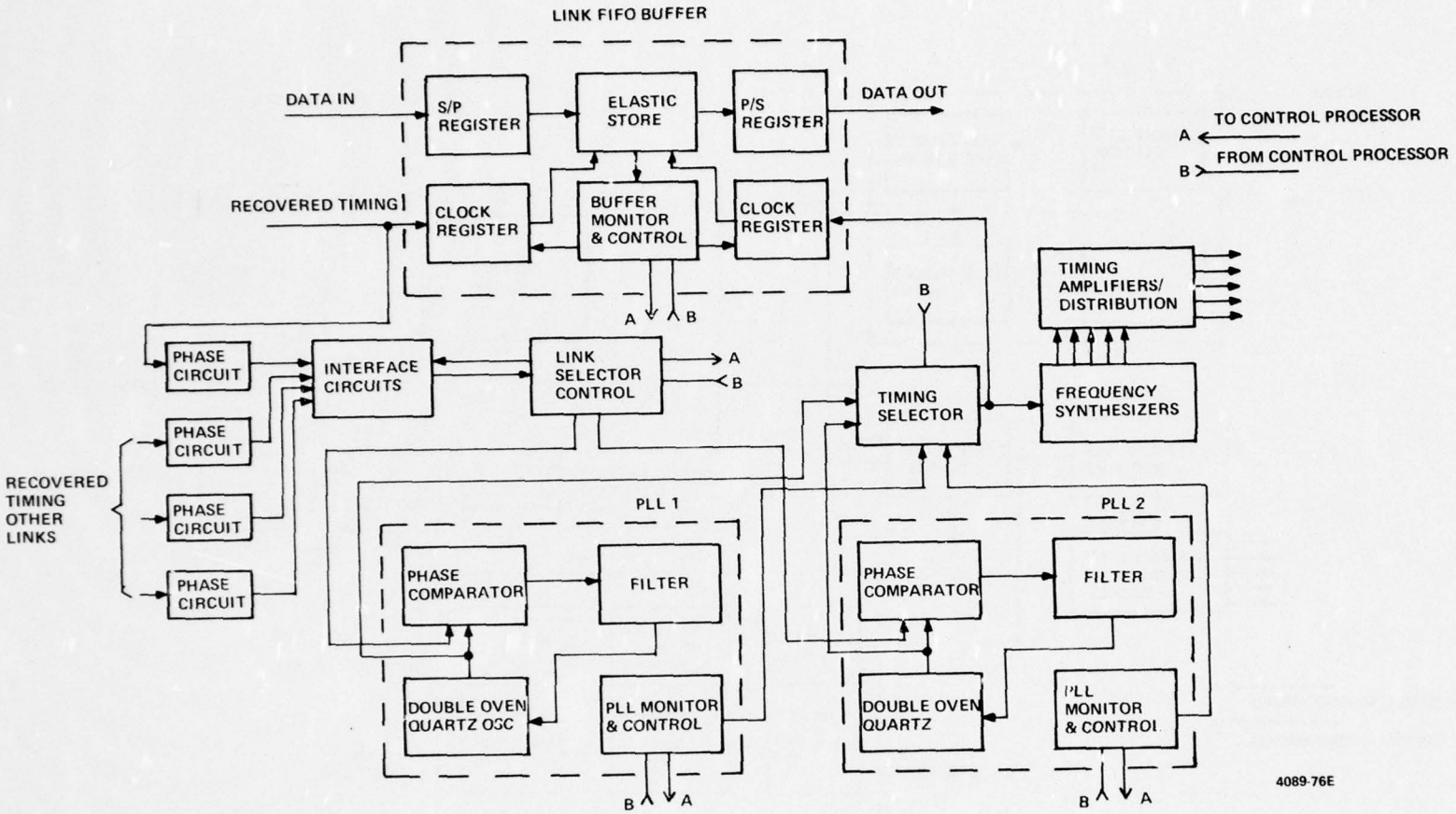


Figure 9-4a. DAX Timing Unit - Slave Configuration

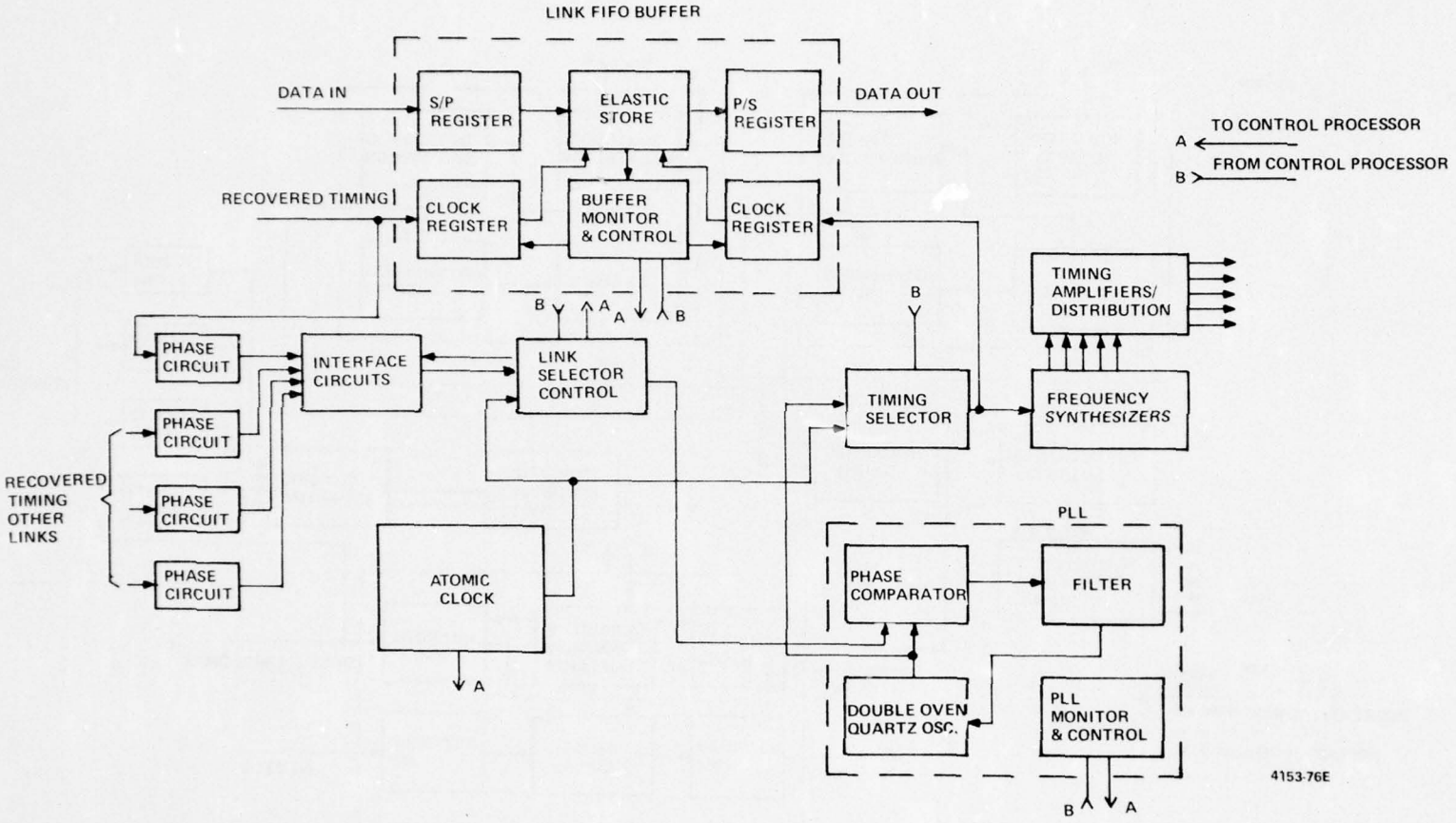


Figure 9-4b. DAX Timing Unit - Master and Standby Master Configuration

Without loss of generality, $\theta(t)$ can be set to zero; however, $\Phi(t)$ cannot be set to zero and a discussion of its behavior will consume the rest of this section.

The instantaneous phase of the oscillator is given by

$$\Omega(t) = 2\pi f_o t + \Phi(t)$$

and the instantaneous spectral frequency by

$$f(t) = \frac{1}{2\pi}\frac{d}{dt}(\Omega(t)) = f_o + \frac{1}{2\pi}\dot{\Phi}(t)$$

The average frequency of the oscillator during a period extending from t_k to $(t_k + \tau)$ is given by

$$\overline{f(t_k)}_\tau = \frac{1}{\tau}\int_{t_k}^{t_k+\tau} f(t)dt = f_o + \frac{1}{2\pi}\frac{\Phi(t_k+\tau) - \Phi(t_k)}{\tau}$$

In general, $\overline{f(t_k)}$ will vary depending upon the length of the averaging period and the starting point t_k. Therefore, such a definition is of little practical value. For this reason, the commonly used definition of stability is the Allan Variance which estimates the expected value of the variance of the frequency fluctuations of an oscillator. Specifically,

if

$$y(t) = \frac{\dot{\Phi}(t)}{2\pi f_o}$$

and

$$\bar{y}_k + \frac{1}{\tau}\int_{t_k}^{t_k+\tau} y(t)\,dt$$

then the Allan Variance is given by

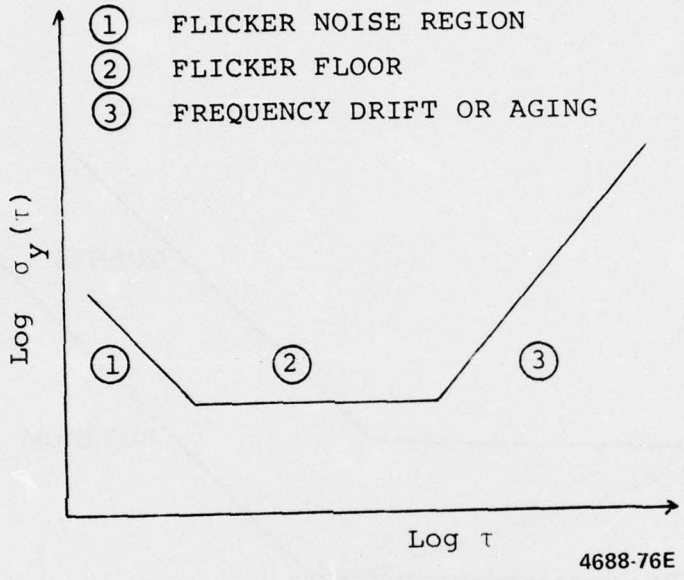
$$\sigma_y^2(\tau) = \left\langle \frac{(\bar{y}_{k+1} - \bar{y}_k)}{2} \right\rangle$$

where $\langle\cdot\rangle$ denotes the infinite time average and $t_{k+1} = t_k + \tau$. In general, the Allan variance provides an invariant measure of the short-term stability of an oscillator. It also characterizes the long-term stability as long as $y(t)$

can be considered stationary over the longer averaging period. Herein after, it is assumed that $\sigma_y^2(\tau)$ is valid at least for τ less than or equal to a year.

Figure 9-5 illustrates the characteristic behavior of $\sigma_y(\tau)$ for the high precision oscillators being considered, namely, double oven quartz crystals, rubidium vapor cells and cesium beam tubes. The shape of the characteristic always exhibits three distinct regions. The first or flicker noise region shows $\sigma_y(\tau)$ decreasing monotonically with τ. The second or flicker floor region shows $\sigma_y(\tau)$ independent of τ. The last region has $\sigma_y(\tau)$ increasing monotonically with τ and corresponds to pure frequency drift or aging. It is interesting to note that as a general rule, crystal clocks exhibit increasing mean frequency with age whereas atomic clocks can either increase or decrease in frequency with age. The values of τ at which the breakpoints occur in Figure 9-5 depend on the type of clock being considered. Figure 9-6 illustrates typical characteristics for commercially available clocks. The flicker region has been omitted for both rubidium and quartz since it is not of interest to the problem at hand. The excellent long-term stability of the cesium clock makes obvious the reason it is used as a primary standard. Based on the curves in Figure 9-6, Table 9-1 provides the assumed time varying frequency characteristics of the three types of clocks being considered. These equations are derived by performing a linear regression fit to the aging region shown in Figure 9-6 for rubidium and quartz and by assuming a flat characteristic for cesium at a poorer than typical long-term stability of 10^{-12} (a worst case assumption).

9.1.3.2.2.2 Link FIFO Buffer - The receive side of each DAX link will be equipped with a link FIFO buffer subsystem. Its primary purpose is to absorb any bit slippage caused by an average frequency differential between recovered timing and locally generated timing. For the type of network synchronization being implemented, the average fill of a link buffer during normal operating conditions is one-half. As shown in Figures 9-4a and 9-4b, the link buffer subsystem is reset to the half full position by inhibiting, via the appropriate clock register, either recovered or nodal timing. The former being used to prevent overflow and the latter to prevent underflow; in either case though, resetting the link buffer causes loss of bit integrity on that link. The link buffer subsystem can also be used to advance or retard the received bit stream in one bit increments. This capability is useful for frame synchronization.



$$v(t) = (A + \theta(t)) \cos (2\pi f_o t + \Phi(t))$$

$$y(t) = \dot{\Phi}(t)/2\pi f_o$$

$$\bar{y}_k = \frac{1}{\tau}\int_{\tau_k}^{\tau_k + \tau} y(t)dt$$

$$\sigma_y^2(\tau) = \left\langle \frac{(\bar{y}_{k+1} - \bar{y}_k)}{2} \right\rangle$$

Figure 9-5. Typical Frequency Stability Characteristic

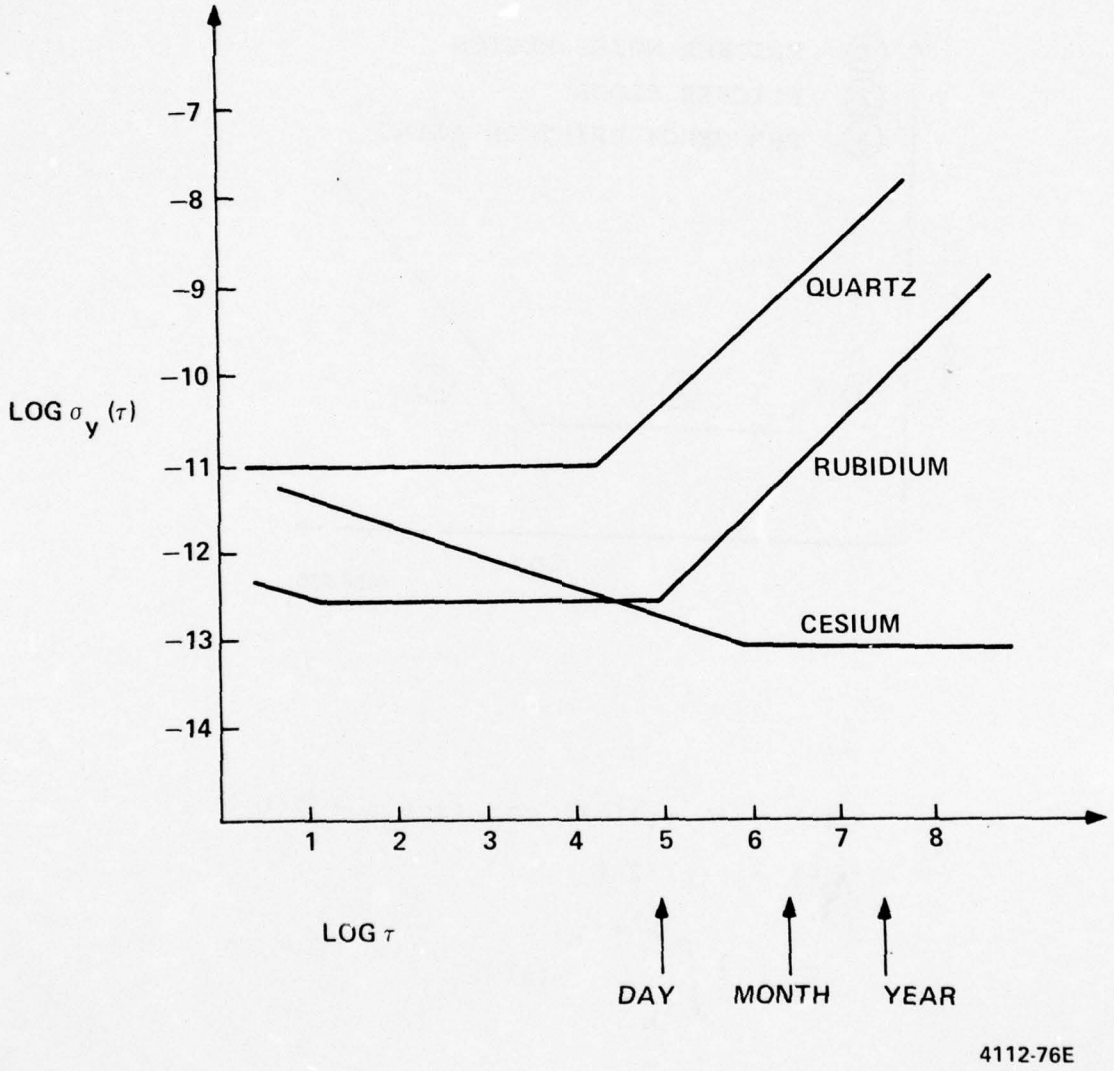


Figure 9-6. High Precision Clock Performance

TABLE 9-1. CLOCK FREQUENCY CHARACTERISTICS

	CLOCK TYPE	CLOCK FREQUENCY (f_o = NOMINAL FREQUENCY)	
1.	Quartz Crystal	$f_o\ (1 \pm 10^{-11})$	$0 \leq t \leq 2 \times 10^4$ (seconds)
		$f_o\ (1 + 1.0378 \times 10^{-15} t^{.9263})$	$2 \times 10^4 \leq t$
2.	Rubidium Vapor Cell	$f_o\ (1 \pm 3.162 \times 13^{-10})$	$0 \leq t \leq 10^5$
		$f_o\ (1 \pm 3.162 \times 10^{-18} t)$	$10^5 \leq t$
3.	Cesium beam tube	$f_o\ (1 \pm 10^{-12})$	$0 \leq t$

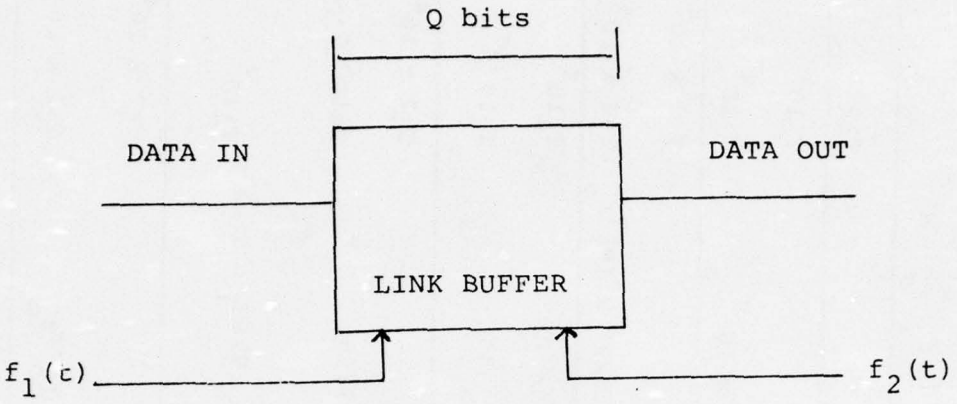
Each link buffer subsystem will include hardware (Buffer Monitor and Control block in Figure 9-4) to monitor and set an output flag for both an overflow/underflow condition and an immiment overflow/underflow condition (e.g., 1/8 or 7/8 full). The outputs of this hardware will be used by the controlling processor to ascertain the status of the link buffer. The input data will undergo serial-to-parallel and parallel-to-serial conversion as it enters and leaves the link buffer, respectively. This is done so that the elastic store can operate in a parallel mode and consequently at a reduced clock rate.

The size of a link buffer will be determined as a function of time to loss of bit integrity on a link between two nodes operating with independent nodal clocks (i.e., the clocks are not slaved to either the master or a local master). The governing equation for this event is given in Figure 9-7a. This equation simply measures the amount of bit slippage between two bit streams with different time varying frequencies f_1 (t) and f_2 (t). It is assumed that f_1 (t) and f_2 (t) age in different directions so that buffer length is determined for a worst case situation. The factor of 2 in the buffer length equation derives from the fact

that bit slippage can be due to an overflow or underflow condition. Application of clock frequency characteristics given in Table 9-1 to the buffer length equation given in Figure 9-7a results in the buffer requirements shown in Figure 9-7b. It is assumed that all link phase locked loops track with a maximum fractional offset of 10^{-10} and that the link bit rate is 1.544×10^{6} b/s. Figure 9-8 shows the buffer length required as a function of the time to loss of bit integrity. The parametric curves provide for any combination of nodal clocks on a link. To illustrate the use of these curves, assume a link buffer length of 500 bits and two nodal clocks free running from an initial offset of 1.544×10^{-4} Hz. Then Table 9-2 specifies the time to first loss of bit integrity for various combinations of nodal clocks. It should be recalled that these results only consider nodal clock instability. Secondary effects such as repeater jitter and transmission delay variations also impact slightly on the required buffer length as described in Section 4.4.

TABLE 9-2. TIME TO FIRST LOSS OF BIT INTEGRITY FOR VARIOUS COMBINATIONS OF NODAL CLOCKS

NODAL CLOCK TYPE		APPROXIMATE TIME TO FIRST LOSS OF BIT INTEGRITY
NODE 1	NODE 2	
Cesium	Cesium	18.3 days
Cesium	Rubidium	18.3 days
Cesium	Quartz	8.2 days
Rubidium	Rubidium	18.3 days
Rubidium	Quartz	8.2 days
Quartz	Quartz	4.1 days



$$Q = 2\int_{o}^{T} |f_1(t) - f_2(t)|dt$$

4689-76E

$$= 2f_o f' T + 2\int_{o}^{T} |f_1'(t) - f_2'(t)|dt$$

where

f_o: nominal frequency

$f_o f'$: initial frequency offset between f_1 and f_2 at start of free-running mode

t_a: time at start of free-running mode (buffer fill at $t_a = 1/2$)

t_b: time at buffer overflow/underflow (loss of bit integrity)

T: $t_b - t_a$

$f_1(t)$: recovered frequency relative to t_a

$f_2(t)$: nodal clock frequency relative to t_a

$f_1'(t)$: recovered frequency relative to t_a and f_o

$f_2'(t)$: nodal clock frequency relative to t_a and f_o

NOTE: It is assumed that f_1 and f_2 drift in opposite directions.

Figure 9-7a. Link Buffer Size Equations

NODAL CLOCK TYPE		LINK BUFFER LENGTH
NODE 1	NODE 2	
Cesium	Cesium	$Q = 2f_oT\ (f' + 2 \times 10^{-12})$ $\quad T \geq 0$
Cesium	Rubidium	$Q = \begin{cases} 2f_oT\ (f' + 1.3162 \times 10^{-12}) & T \leq 10^5 \\ 2f_oT\ (f' + 10^{-12} + 1.5812 \times 10^{-18}T) + 3.1623 \times 10^{-8}f_o & T \geq 10^5 \end{cases}$
Cesium	Quartz	$Q = \begin{cases} 2f_oT\ (f' + 1.1 \times 10^{-11}) & T \leq 2 \times 10^4 \\ 2f_oT\ (f' + 10^{-12} + 5.3875 \times 10^{-16}T^{.9263}) + 1.9227 \times 10^{-7}f_o & T \geq 2 \times 10^4 \end{cases}$
Rubidium	Rubidium	$Q = \begin{cases} 2f_oT\ (f' + 6.3246 \times 10^{-13}) & T \leq 10^5 \\ 2f_oT\ (f' + 3.1623 \times 10^{-18}T) + 6.3246 \times 10^{-8}f_o & T \geq 10^5 \end{cases}$
Rubidium	Quartz	$Q = \begin{cases} 2f_oT\ (f' + 1.0316 \times 10^{-11}) & T \leq 2 \times 10^4 \\ 2f_oT\ (f' + 3.1623 \times 10^{-13} + 5.3875 \times 10^{-16}T^{.9263}) + 1.9227 \times 10^{-7}f_o & 2\times10^4 \leq T \leq 10^5 \\ 2f_oT\ (f' + 1.58115 \times 10^{-18}T + 5.3875 \times 10^{-16}T^{.9263}) + 2.2391 \times 10^{-7}f_o & T \geq 10^5 \end{cases}$
Quartz	Quartz	$Q = \begin{cases} 2f_oT\ (f' + 2 \times 10^{-11}) & T \leq 2 \times 10^4 \\ 2f_oT\ (f' + 1.0775 \times 10^{-15}T^{.9263}) + 3.8455 \times 10^{-7}f_o & T \geq 2 \times 10 \end{cases}$

4687-76E

Figure 9-7b. Link Buffer Size Equations

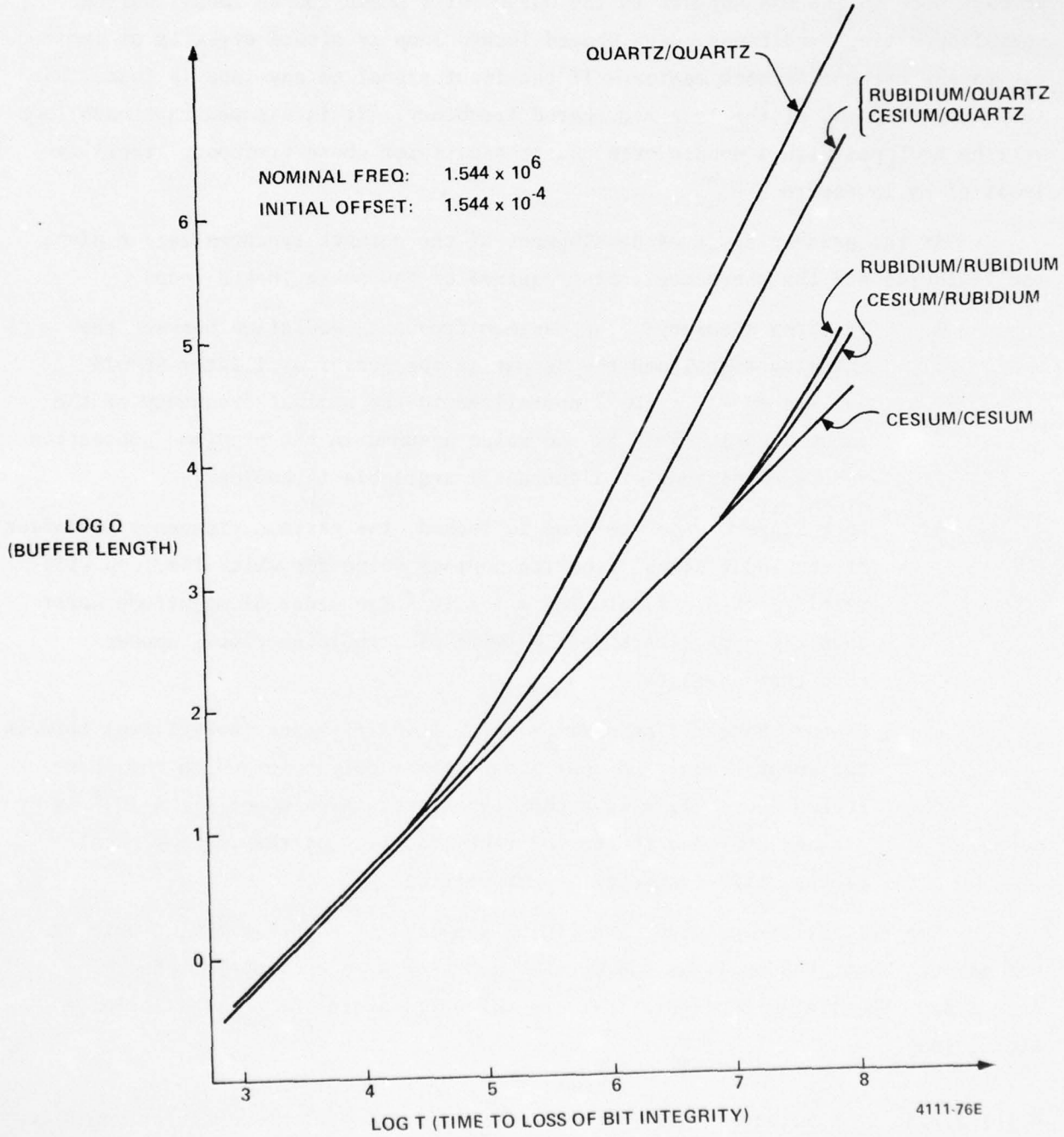


Figure 9-8. DAX Buffer Requirements

AD-A039 549 GTE SYLVANIA INC NEEDHAM HEIGHTS MASS ELECTRONIC SYS--ETC F/G 17/2
SENET-DAX STUDY. VOLUME 2.(U)
JUN 76 DCA100-75-C-0071
UNCLASSIFIED FR76-1-VOL-2 NL

2 OF 3
AD A039549

9.1.3.2.2.3 Phase Locked Loop - As shown in Figure 9-4, the primary timing source at each node in the DAX network is the output of a phase locked loop. During normal operating conditions, each phased locked loop is slaved directly or indirectly to the current network master. If the input signal to any loop is lost, that loop will free run at the last remembered frequency. It is assumed that each loop will be equipped with a double oven quartz oscillator whose frequency stability is as given in Figure 9-6.

At the present stage of development of the network synchronization plan, the following are the characteristics required of the phase locked loop:

a. Tracking accuracy: The maximum frequency deviation between the incoming signal and the output of the quartz oscillator should not exceed $\pm$ 1 x 10^{-10} normalized to the nominal frequency of the input signal. This is the value assumed in the previous subsection and is consistent with currently available technology.

b. Lock Range: Once the loop is locked, the maximum frequency deviation of the input signal from its nominal value for which the loop will remain locked. A value of $\pm$ 1 x 10^{-9} (an order of magnitude worst than the approximate yearly drift of a rubidium clock) appears more than adequate

c. Capture Range: The maximum frequency difference (normalized) between the input signal and quartz oscillator output for which the phase locked loop will always come into lock. A value of $\pm$ 1 x 10^{-9} is assumed, but due to the inherent stability of the various nodal clocks, this parameter is not critical.

For reliability purpose, the timing supplies of each DAX timing unit are redundant. Table 9-3 explains the primary and backup mode of operations by node type. The timing configurations are set up to avoid the possibility of a timing loop.

9.1.3.2.2.4 Link Selector - The inputs to this device are the possible inputs to the phase locked loop. Included in this category are the recovered timing signal from the master, recovered timing signals from the various standby masters, and

TABLE 9-3. NODAL TIMING CONFIGURATIONS

NODE TYPE	PRIMARY TIMING	BACKUP TIMING
1. Master	PLL slaved to colocated Cesium atomic clock	1. PLL slaved to standby master 2. Cesium atomic clock 3. Free running PLL
2. Local Master/ Standby Master	PLL slaved to master	1. PLL slaved to colocated Rubidium atomic clock 2. PLL slaved to another standby master 3. Free running PLL
3. Local Master	PLL-1 slaved to master	1. PLL-1 slaved to standby master 2. PLL-2 slaved to master 3. PLL-2 slaved to stnadby master 4. Free running PLL-1 5. Free running PLL-2
4. Slave	PLL-1 slaved to local master	1. PLL-2 slaved to local master 2. PLL-1 free running 3. PLL-2 free running

PLL: Phase Locked Loop

and the output of any colocated atomic clock. The device must be capable of selecting the highest precedence input signal which is providing valid timing. If full versatility is desired, the device must be programmable by the control processor. Obviously, the link selector must include sufficient hardware/software to detect which of its inputs are providing valid timing signals.

9.1.3.2.2.5 Timing Selector - This hardware would perform the switching between the possible timing configurations indicated in Table 9-3. The device would switch inputs only on command of the control processor.

9.1.3.2.2.6 Phase Circuit/Interface Circuit - To avoid large phase hits when switching inputs to the phase locked loops, a phase adjusting circuit is provided for each possible input signal. The interface circuit is included to reconcile any differences between the operating frequency of the phase locked loops and the transmission rate (i.e., it performs any frequency conversions required).

9.1.3.2.2.7 Timing Distribution/Frequency Synthesizer - The output of the phase locked loop is a highly accurate and stable single frequency signal. This equipment generates from the phase locked loop output the various frequencies and amplitudes required by DAX switching and transmission subsystems.

9.1.3.2.2.8 Control Processor - As designed, the control processor plays a very small role in the overall operation of the network synchronization system. Its functions are essentially limited to the following:

a. Setting the precedence of the input signals to the link selector

b. Monitoring phase locked loops and the atomic clock (if there is one) status so that proper information can be fed to the timing selector

c. Monitoring all link buffers and commanding corrective actions (buffer reset, advance one bit, etc.) when necessary

d. Processing of status information for record keeping and maintenance purposes.

9.1.3.3 Error Control

A 16-bit frame check sequence (FCS) for Class II data packet error control is recommended in Section 4.6. The placement of the FCS in the Class II region is illustrated in Figure 9-9. Industry requirements for a Cyclic Redundancy Check Character (CRCC) generator/checker have prompted the production of a polynomial generator chip MC8506P (Figure 9-10) which can be utilized in this application.

During data transmission, the chip divides n data bits by the CRCC-CCITT polynomial ($X^{16} + X^{12} + X^5 + 1$). The 16-bit remainder is inverted and appended to the data stream.

At the receiver the division proceeds through n + 16 bits. If bit integrity was maintained in transmission, the chip provides a Pattern Match output signal.

Future studies of this subject may require a larger FCS. This would be possible in 4-bit increments using multiple MC8504P Universal Presettable Polynomial Generator Chips and external control gates.

9.2 MASTER FRAME SYNCHRONIZATION

9.2.1 Problem

In each master frame, the location of each virtual circuit switched connection is only known relative to the starting position of the master frame. Therefore, before Class I information can be transmitted over an inter-DAX link, master frame synchronization must be established and maintained. The method by which the DAX accomplishes this type of synchronization and the hardware/software implementation of this method are the subject of this section. It is assumed that bit synchronization as described in Section 9.1, has already been achieved.

9.2.2 Objectives

In Section 4.2 a general description of the chosen master frame synchronization method is provided. Using this as a starting point, the objectives of this section are as follows:

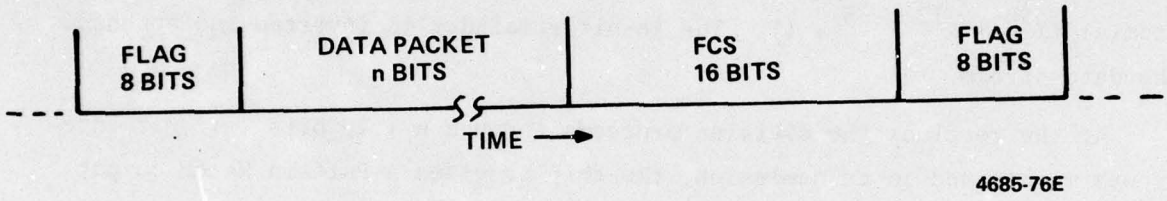


Figure 9-9. Random Data

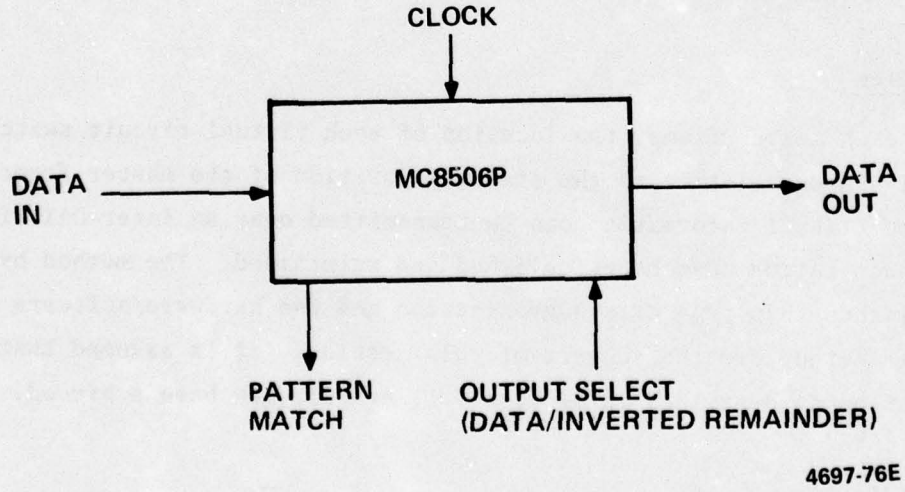


Figure 9-10. Polynomial Generator/Checker

a. Finalize the synchronization method specified in Section 4.2

b. Examine the tradeoff between transmission efficiency and synchronization performance alluded to in Section 4.2

c. Minimize, to the maximum extent possible, the software requirements of the frame maintenance unit (FMU). This objective is consistent with the basic SENET-DAX conceptual design philosophy.

9.2.3 Analysis and Results

9.2.3.1 System Description

The master frame synchronization process, which is examined in detail in subsequent subsections, is essentially that outlined in Section 4.2. Specifically, master frame synchronization of each link at each DAX is controlled by an FMU. This device maintains synchronization in the receive direction and assists its companion FMU at the remote DAX in maintaining synchronization in its receive direction. Three start-of-frame markers SOF-1, SOF-2 and SOF-3 are utilized by the FMU for this purpose. The conditions under which each SOF is employed are described in Table 9-4. Note that SOF-2 is the binary complement of SOF-1 and that SOF-3 is the sequence formed by concatenating the three sequences SOF-1, SOF-1, and SOF-2. The reasons for choosing SOF-1 to be the 16-bit sequence illustrated in Table 9-4 and for SOF-2 and SOF-3 to be derivable from SOF-1 are discussed subsequently.

The FMU has two operational states or modes, a frame maintenance mode and a frame acquisition mode. When operating in the frame maintenance mode, the FMU assumes that the link is in-sync and monitors for loss of this condition. To ensure that the FMU performs satisfactorily in this mode, the algorithm by which it chooses between in-sync and out-of-sync must be such as to simultaneously minimize or make negligible the following probabilities.

a. Prob (false out-of-sync alarm), P_{FOA}: this denotes the probability of the FMU indicating out-of-sync when in fact the link is in-sync

b. Prob (loss-of-sync miss), P_{LM}: this denotes the probability of the FMU indicating in-sync when in fact the link is out-of-sync.

TABLE 9-4. MASTER FRAME CONTENTS AS A FUNCTION OF SYNCHRONIZATION STATE

SYNC STATUS		FMU MODE		TRANSMITTED SIGNALS			
DAX 1	DAX 2	DAX 1	DAX 2	DAX 1		DAX 2	
				Frame	Data	Frame	Data
In-sync	In-sync	Maintenance	Maintenance	SOF-1	Random	SOF-1	Random
In-sync	out-of-sync	Maintenance	Acquisition	SOF-3	Random	SOF-2	Random
Out-of-sync	In-sync	Acquisition	Maintenance	SOF-2	Random	SOF-3	Random
Out-of sync	Out-of-sync	Acquisition	Acquisition	SOF-2	Random	SOF-2	Random

SOF-1 11101011100010000

SOF-2 $\overline{\text{SOF-1}}$

SOF-3 SOF-1 || SOF-1 || SOF-2

Note: $\overline{x}$ Binary complement of x

x||y Sequence x concatenated with sequence y

In the frame acquisition mode, the FMU assumes an out-of-sync condition and scans the input for a valid SOF. This portion of the acquisition mode, that is the scanning of the input bit stream for a valid SOF, is referred to as the search phase. Once the FMU locates what it believes to be the valid synchronization position, it terminates the search phase and enters the check phase. During this next phase, the FMU attempts to verify that the assumed synchronization position is in fact the true synchronization position. If the check phase indicates that the assumed position is incorrect, the FMU returns to the search phase; otherwise, the FMU assumes that resynchronization has been achieved and returns to the maintenance mode. A measure of the performance of the FMU is the acquisition mode is the Prob (false in-sync alarm) of P_{FIA}. This is the probability that acquisition mode locks on to a position which is not the true synchronization position. The acquisition mode must be designed to make this probability negligible. Another factor of interest in the acquisition mode is the average time to acquire synchronization, <u>. This quantity denotes the average number of bits required to lock on to the correct synchronization position and excludes the check phase when the correct synchronization position is found. Since on the average, the FMU must check half the frame before reaching the transmitted SOF, <u> is lower bounded by one-half frame.

9.2.3.2 Definitions and Assumptions

The following definitions are referred to throughout the remainder of Section 9.2 and are listed here for easy reference:

a. N_i = length in bits of SOF-i, i = 1, 2, 3.

b. M = length in bits of the master frame. Useless stated otherwise, M is assumed to be 15440 bits

c. e = probability of a bit error. It is assumed as discussed in assumption (b) below that bit errors are independent events

d. α = the largest integer smaller than or equal to $(M-N_i)/N_i$.

e. τ = maximum number of bit disagreements permitted between the SOF being considered and the assumed synchronization position being observed for which the FMU decides that the observed position contains the true synchronization pattern

f. s = number of master frames used in the check phase of the acquisition mode

g. n = length in master frames of the frame maintenance test by which the FMU determines loss of synchronization

h. t = minimum number of positive correlations required in frame maintenance test in order to declare an in-sync condition

i. $P(A|B)$ = proba-ility that the FMU accepts the position being observed as the true synchronization position conditioned on the fact that it is the true synchronization position. For a particular SOF, this is given by

$$P(A|B) = \sum_{k=o}^{\tau} \binom{N_i}{k} e^k \quad (1-e)^{N_i-k}$$

j. $P(\overline{A}|B)$ = probability that the FMU rejects the position being observed as the true synchronization position conditioned on the fact that it is the true synchronization position. This is given by

$$P(\overline{A}|B) = 1 - P(A|B)$$

k. $P(A|\overline{B})$ = probability that the FMU accepts the position being observed as the true synchronization position conditioned on the fact that it is not the true synchronization position. For random data and a particular SOF, this is given by

$$P(A|\overline{B}) = \sum_{k=o}^{\tau} \binom{N_i}{k} 2^{-N_i}$$

l. P_{RS} = in an errorless environment, the probability that the search phase of the acquisition mode selects an incorrect synchronization position

m. P_{FI} = probability that the first indication of synchronization in the search phase of the acquisition mode is the true synchronization position

n. P_{FIA} = probability that the acquisition mode locks on to a synchronization position that is not the true synchronization position

o. P_{FOA} = probability that the frame maintenance mode indicates loss of synchronization when in fact synchronization has not been lost

p. P_{LM} = probability that the frame maintenance mode fails to detect loss of synchronization.

The following assumptions are made concerning the synchronization process and channel characterization:

a. Master Frame Structure: the master frame is composed of two sections. The first will always be one of three known bit patterns SOF-i (i = 1,2 or 3) whose respective lengths are N_i bits. The second section consists of random data (Class I and Class II information) $M-N_i$ bits long.

b. Noise Model: the frame synchronization process will be designed for an environment in which errors occur randomly with a probability of 1×10^{-2}. It should be noted that this represents a worst-case time weighted average of the error environment postulated in Section 4.2. Although the synchronization process is designed for a purely random noise environment, it performs adequately in the burst environment. As discussed in Section 9.2.3.7, the FMU is able to maintain synchronization during a burst, but may be forced to ride out a burst if it is attempting to acquire synchronization. This would entail a maximum increase in <u> of 50 msec, which is not considered excessive.

c. In the acquisition mode, the true syncrhonization position can start in any of M equiprobable and independent positions

d. The check phase is such that it will always reject a false synchronization position in "s" frames

e. The system is defined to be in-sync the first time the FMU accepts the true synchronization position.

These assumptions entail no loss of generality and are consistent with the system description above.

9.2.3.3 SOF Bit Pattern Considerations

A valid SOF can appear in any of three regions within the master frame: the true synchronization position; a region comprised partly of the true synchronization position and partly of random data, denoted as the overlap region; and the pure-random data region. In the frame maintenance mode, the FMU scans only region 1 while testing for valid SOF's. Clearly, then, for a given SOF length, the probability of obtaining a false synchronization pattern correlation (τ or fewer errors) is independent of the particular bit pattern used and depends only on the error environment. In the search mode, the FMU sequentially scans all these regions searching for a valid SOF. For regions 1 and 3, the same conclusion applies; namely, for a given SOF length, the bit pattern used plays no role in determining the probability of obtaining a false synchronization pattern correlation. However, the SOF bit pattern does impact on the probability of a false correlation in the overlap region since this probability is a function of the number of bits in agreement with the true SOF. To minimize the impact of SOF simulations in region 2, it is necessary to satisfy the following bit pattern restriction,

$$(\rho_o, \rho_1, \rho_2, \cdots, \rho_{j-1}) \neq (\rho_{N_i-1}, \rho_{N_i-j+1}), \cdots, \rho_{N_i-1})$$

$$j = 1, 2, \ldots, N_i$$

where $(\rho_o, \rho_1, \cdots, \rho_{N_i-1})$ = SOF- i.

This ensures that in an error free environment there can be no simulation of the SOF in region 2 and also serves to minimize the probability of this event in a noisy environment. It is interesting to note that neither the five bit nor thirteen bit Barker sequences satisfy this requirement. As a result, both are ruled out as possible SOFs. As a consequence of the above SOF bit pattern restriction and the fact that the acquisition mode includes a check phase, it is assumed that P_{FIA} is calculable by considering only region 3. That is, the contribution to P_{FIA} due to the

possibility of a simulated SOF in region 2 is negligible compared to the same possibility for region 3.

Due to the assumed length of SOF-3, it is not necessary that it strictly adhere to the above bit pattern restriction. In fact, as long as it is of reasonable length (e.g., $\geq$ 25 bits), SOF-3 can be chosen for convenience rather than as a result of careful analysis. Thus, if SOF-2 is chosen to be the binary complement of SOF-2 (thereby maximizing the Hamming distance between the two) then SOF-3 can be the convenient pattern SOF-1 ||SOF-1 || SOF-2. Such a choice for SOF-3 would simplify the hardware implementation of the FMU and the transition of the FMU from the acquisition mode to the maintenance mode.

9.2.3.4 Frame Maintenance Mode

The FMU enters the frame maintenance mode upon acquisition of frame synchronization and maintains this mode until it determines that synchronization has been lost. There are numerous methods by which the FMU can decide between an in-sync and out-of-sync condition. The method to be used here is an n-frame test using non-overlapping frames and a fixed threshold decision process. The primary reasons for selecting this method are as follows:

a. The implementation is straightforward and is easily realized with hardware or software

b. The implementation provides excellent performance for carefully chosen values of n, the fixed threshold, and other test parameters.

The specific operations of the chosen technique require that for each master frame for n successive master frames, the FMU correlate (compares bit-by-bit) the bit pattern in the assumed synchronization position with the expected SOF. If a correlation results in τ or fewer bit disagreements, a positive correlation is recorded; otherwise, a negative correlation is recorded. Whereupon, if in n such comparisons the number of positive correlations equals or exceeds some fixed threshold, an in-sync condition is declared and the test is repeated with the next n master frames. If the number is less than this threshold an out-of-sync condition is declared, the frame maintenance mode is terminated, and the FMU initiates the frame acquisition mode.

An alternative to using n new master frames each test is to perform the test with each new master frame by using that frame and last n-1 frames. This alternate approach would be implemented as an n-frame window through which the incoming bit stream is shifted one frame at a time. The primary advantage of this approach over the non-overlapping method is that it reduces the mean time to determine loss of synchronization. Its primary disadvantage is that it utilizes extensive memory which could be detrimental in a burst noise environment. For example, as described in Section 4.2, it is postulated that a burst could last for up to 50 msec or 5 master frame periods. Thus, the burst in whole or part would effect n + 5 successive tests in the window approach but would only effect at most two successive tests (for n>4, which is likely) in the non-overlapping approach. Because of the impact an extended burst would have on P_{FOA}, the window approach will not be used.

As described in Section 9.2.3.1, the performance of the frame maintenance mode is measured in terms of P_{FOA} and P_{LM}. These probabilities, as may be expected, are intimately related to n, N_i, t, and test threshold value. Unfortunately though, P_{FOA} and P_{LM} having opposing requirements, that is, as the test parameters are varied to decrease (increase) P_{FOA}, P_{LM} increases (decreases). A full parametric analysis to jointly optimize P_{FLA} and P_{LM} would require extensive computer time. In order to avoid this necessity, a value of n will be chosen which results in a reasonable mean time to determine loss of synchronization. For this value of n, the test threshold will be parametrically varied for likely N_i candidates to determine if reasonable values of P_{FOA} and P_{LM} result. If not, the process will be repeated for different values of n. Although this procedure may not result in an optimal solution, it does provide excellent results.

As a starting point, the following is assumed

$n = 10$ master frames

$N_i = 13$ or 16 bits

$\tau = 1$

As is customary in the maintenance mode, τ is chosen to be approximately equal to its mean plus three standard deviations ($N_i e + 3(N_i e(1-e)^{.5})$. P_{FOA}, in terms of previously defined parameters, is given by

$$P_{FOA} = \sum_{k=n-t+1}^{n} \binom{n}{k} (P(\bar{A}|B))^{k} \quad (P(A|B))^{n-k}$$

where $t \triangleq$ test threshold,

$$P(A|B) = \sum_{k=o}^{t} \binom{N_i}{k} e^{k} \quad (1-e)^{N_i-k} \qquad \text{and}$$

$$P(\bar{A}|B) = 1- \quad P(A|B)$$

A determination of P_{LM} depends on what region of the master frame the FMU is monitoring. As discussed previously, a worst case assumption is to assume that the FMU is monitoring the random data region. In which case

$$P_{lm} \approx \sum_{k=t}^{n} \binom{n}{k} (P(A|\bar{B}))^{k} \quad (1- P(A|\bar{B}))^{n-k}$$

$$P(A|\bar{B}) = \sum_{k=o}^{\tau} \binom{N_i}{k} \quad 2^{-N_i}$$

Figure 9-11 illustrates the result of the P_{FOA} and P_{LM} calculations for the specified parameter values. Table 9-5 summarizes from Figure 1 the best results

TABLE 9-5. PERFORMANCE OF FRAME MAINTENANCE UNIT

	N_i (N_1 or N_2)	
	13 bits	16 bits
P_{FOA}	3×10^{-11} (t = 5)	2.2×10^{-12} (t = 4)
P_{LM}	3.6×10^{-12} (t = 5)	9.5×10^{-13} (t = 4)

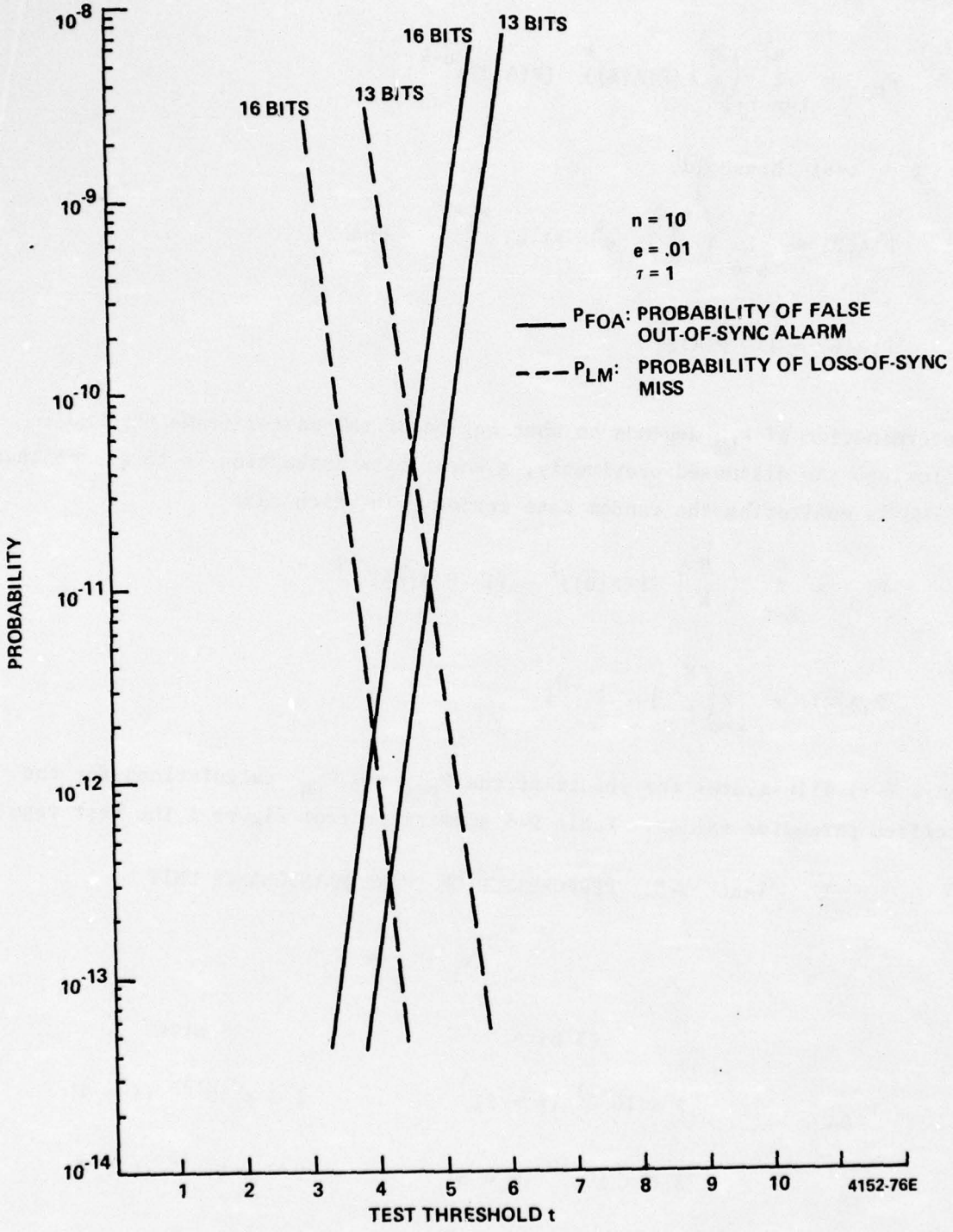


Figure 9-11. P_{FOA} and P_{LM} as a Function of Test Threshold With N_i as a Parameter

obtainable assuming P_{FOA} and P_{LM} are equally weighted. A consideration of Table 9-5 reveals that both the 13-bit and 16-bit SOF provide performance in the range that would be required. Although the edge in performance goes to the 16-bit length, this edge is gained at the cost of transmission efficiency since the 16-bit length requires transmitting an additional 3 bits per frame. A decision as to which SOF length is preferable will be deferred until the impact of frame acquisition has been considered.

9.2.3.5 Frame Acquisition Mode

The operation of the FMU during frame acquisition is a 2-phase process as described in Section 9.2.3.1. It should be recalled that the FMU must be capable of synchronizing with SOF-2 or SOF-3. Since SOF-2 is the shorter of two sequences, the frame acquisition mode must be designed to perform adequately with SOF-2. However, SOF-2 is also used in the frame maintenance mode; thus, its length must be such as to satisfy the requirements of both modes.

The principal parameters which control the performance of the frame acquisition mode are τ, s and N_i, where performance is measured in terms of $\langle u \rangle$ and P_{FIA}. It can be shown (45) that subject to the assumptions listed in Section 9.2.3.2, the average time to acquire synchronization in the acquisition mode is

$$\langle u \rangle = \frac{M+1}{2} + M \frac{P(A|B)}{P(A|B)} + \frac{M(M-1)}{2} P(A|\bar{B}) \left(1 + 2 \frac{P(\bar{A}|B)}{P(A|B)}\right) s$$

Observe that if in the search phase the probability of accepting a false synchronization position ($P(A|\bar{B})$) is large, then $\langle u \rangle$ is proportional to the square of the frame length. For this reason it is important to minimize $P(A|\bar{B})$. A related parameter of interest is the variance of u. However, this is an extremely difficult quantity to calculate. In its place, the parameter P_{FI} will be calculated. This is the probability that the first indication of frame synchronization in the search phase is the true synchronization position. As shown in [1], this is given by

$$P_{FI} = \frac{1}{M} \frac{P(A|B)}{P(A|\bar{B})} \quad 1 \frac{1 - (1 - P(A|\bar{B}))^M}{- P(\bar{A}|B) \quad (1-P(A|\bar{B}))^M}$$

The remaining performance indicator of interest is P_{FIA}. For reasons discussed previously, this probability will be derived by neglecting the contribution of the overlap region of the master frame. It can be shown [48] that the probability of choosing a simulated SOF in the random data region instead of the true synchronization position (P_{RS}) is given by

$$P_{rs} = \sum_{k=1}^{\alpha} (-1)^{k+1} \frac{1}{k+1} \binom{M-N_i-(N_i-1)k}{k} 2^{-N_i k}$$

where it has been assumed that $\tau = o$ (based on the mean number of bit errors expected). Therefore, if the check phase requires s master frames for verification of synchronization then P_{FIA} is approximately given by

$$P_{FIA} = 2^{-N_i s} P_{RS}$$

Table 9-6 provides an evaluation of the performance of the acquisition mode for different parameters of interest. As may be seen there, in order to obtain tolerable values for P_{FIA} when synchronizing with SOF-2, two frames are required in the check phase for both the 13- and 16-bit lengths. However, a single check frame is adequate when synchronizing with N_3.

At this point it will be possible to select a length for SOF-1 which, in turn, determines N_2 and N_3. As a result of the frame maintenance analysis, it was seen that both the 13- and 16-bit lengths provide excellent performance. However, with regard to frame acquisition the 16-bit length decisively outperforms the 13-bit length. As a result, a length of 16 bits is proposed for N_1. A choice of a particular bit pattern for SOF-1 is not critical but must be consistent with results of Section 9.2.3.3. In [39] it is shown that the 16-bit pattern given in Table 9-6 provides excellent performance and is, in fact, the optimal pattern subject to the assumptions listed in [39]. It is proposed, therefore, that this bit pattern be used for SOF-1.

TABLE 9-6. PERFORMANCE OF THE FRAME ACQUISITION MODE

					s = 1		s = 2		
SOF	$P(A\|B)$	$P(\bar{A}\|B)$	$P(A\|\bar{B})$	<u> (frames)	P_{FI}	P_{FIA}	<u> (frames)	P_{FI}	P_{FIA}
N_2 = 13 bits	.878	.122	1.22×10^{-4}	1.84	.274	6.65×10^{-5}	3.05	.274	8.11×10^{-9}
N_2 = 16 bits	.851	.149	1.53×10^{-5}	.83	.961	1.66×10^{-6}	.99	.961	2.54×10^{-11}
N_3 = 39 bits	.676	.324	1.82×10^{-12}	.98	~1	2.54×10^{-20}	.98	~1	4.62×10^{-32}
N_3 = 48 bits	.617	.383	3.55×10^{-15}	1.12	~1	9.74×10^{-26}	1.12	~1	3.46×10^{-43}

e = .01

τ = 0

9.2.3.6 FMU Conceptual Design

Figure 9-12 illustrates a block diagram of the FMU based on the frame synchronization concepts discussed so far. The figure identifies 3 correlators, one for each SOF. The SOF-1 correlator is used only in the maintenance mode, SOF-3 only in the acquisition mode, and SOF-2 in both the maintenance and acquisition modes. Because SOF-1 and SOF-2 are binary complements, the hardware implementation of the two respective correlators would probably be a single device with either hardware or software determining which of the two sequences was sent in any frame. Figure 9-12 also identifies two controllers, one for each mode of operation. The controllers use the output of the correlators to maintain synchronization on the receive side of the link and to aid the remote DAX in maintaining synchronization on its receive side of the link. The two controllers drive the interface logic which provides four output flags. Two of the flags would be of interest to the main processor, F_1 and F_4. These flags indicate the synchronization status on the receive side of the link and at the remote DAX, respectively. Flags F_2, F_3, F_4 specify which SOF is to be transmitted to the remote DAX. The actual insertion of an SOF into the transmitted bit streams could be accomplished within the FMU or externally.

The FMU controllers could be special purpose hardware or possibly a microprocessor. In either case, the control algorithms to be implemented are shown in Figures 9-13a and 9-13b. The frame acquisition algorithm provides the capability of synchronizing with either SOF-2 or SOF-3. Based on the acquisition analysis, a two frame check phase is used with SOF-2 and a single frame check phase with SOF-3. The frame maintenance algorithm permits synchronization to be maintained with either SOF-1 or SOF-2. A special counter is employed to count the number of SOF-2s received during each 10 frame test. Observe that if an in-sync condition is declared at the conclusion of a 10 frame test and simultaneously the SOF-2 count exceeds one, the F_4 flag is set (resulting in the transmission of SOF-3). The justification for using a single SOF-2 to indicate an out-of-sync condition at the remote DAX derives from the probability of receiving an SOF-2 conditioned on the fact that an SOF-1 was transmitted. Even in the middle of a burst, this probability is

$$\binom{16}{15} (.2)^{15} (.8)^{1} + (.2)^{16} = 4.26 \times 10^{-10}$$

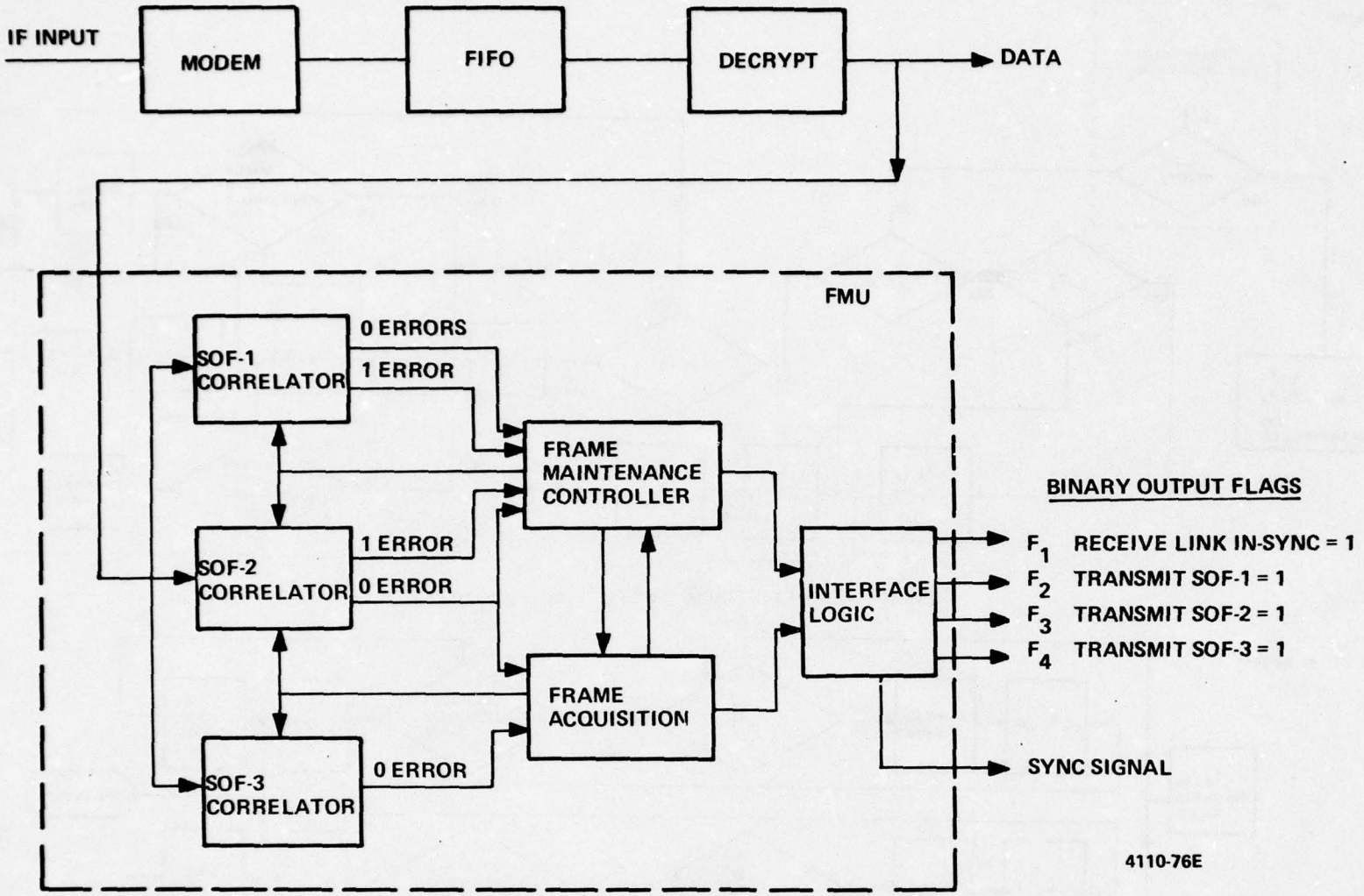


Figure 9-12. Frame Maintenance Unit (FMU)

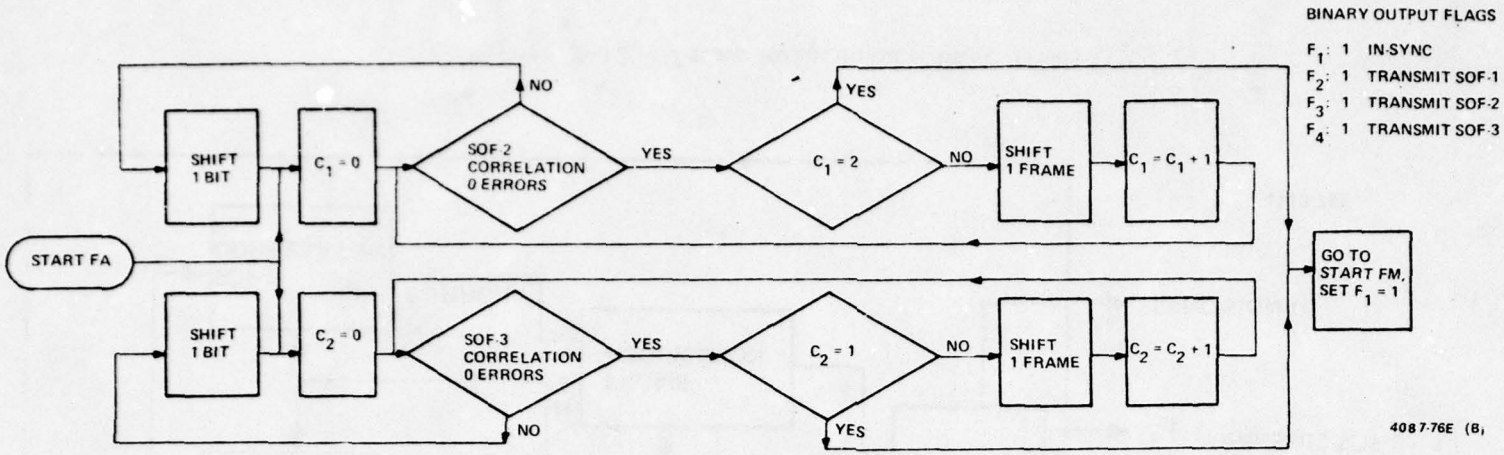


Figure 9-13a. Frame Acquisition Algorithm

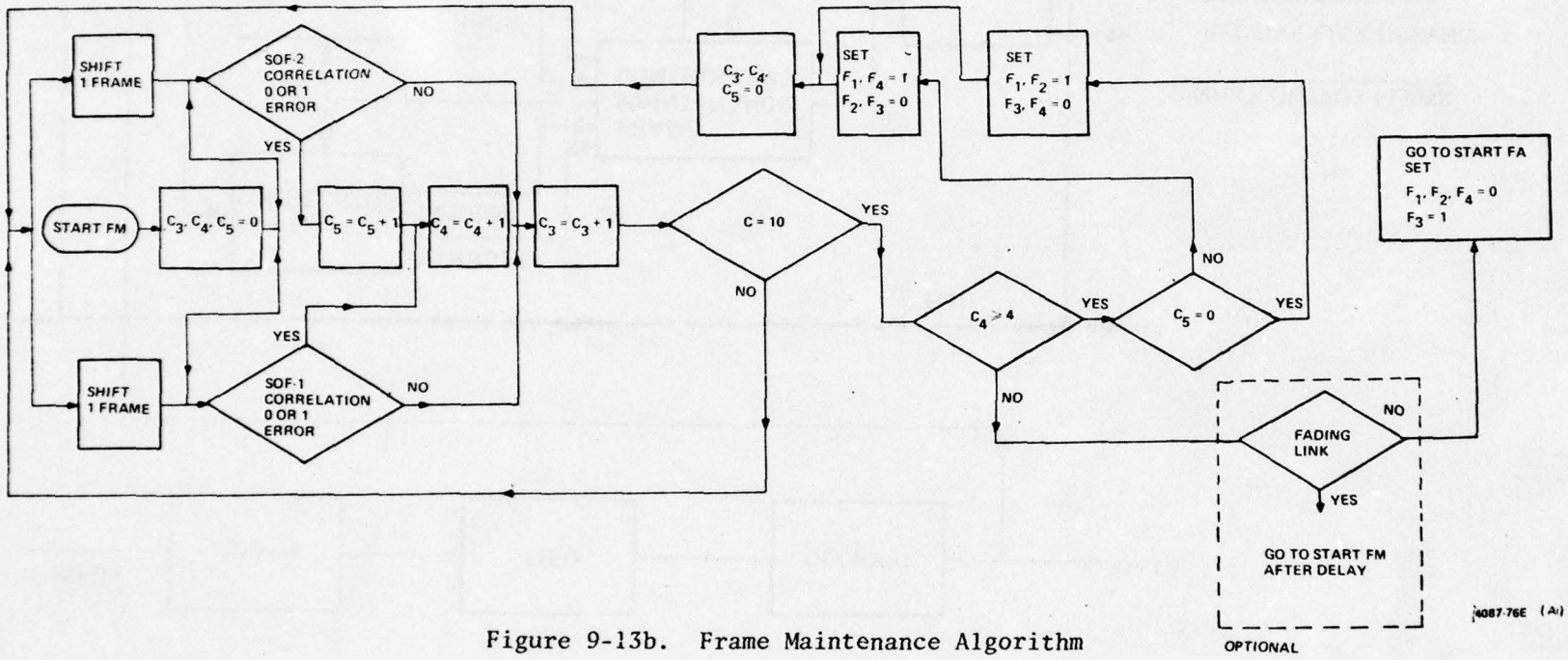


Figure 9-13b. Frame Maintenance Algorithm

As described above, the FMU controllers could be implemented with hardware or software. In all likelihood, the device will be a microprocessor since this would provide the capability of easily altering the various algorithm parameters such as τ, t, s, etc. This versatility would, in turn, permit tailoring each FMU to the channel environment in which it would be used.

9.2.3.7 Burst Errors

The FMU has been designed for a random error environment of 1×10^{-2}. The performance of the FMU in the burst error environment described in Section 4.2 is now addressed. With regard to frame acquisition, the FMU may not be able to synchronize during a burst but should easily be able to synchronize between bursts where the bit error rate is drastically reduced ($e = 1 \times 10^{-3}$). Fortunately though P_{FIA} is not significantly affected by the presence of bursts so there is no additional risk of falsely declaring synchronization during a burst period.

With regard to frame maintenance, it is desirable that the FMU does not falsely declare loss of synchronization during a burst period. The worst case situation here is a 50-msec burst which affects 6 master frames of the 10 frame test. The probability of falsely declaring loss-of-sync is approximately given by

> Prob (one or more negative SOF correlations in the four remaining good frames) x Prob (burst $\geq$ (50-D) msec) x Prob (burst starting within the SOF)

where D is the length of SOF in msec and it is assumed that all of the events are independent. If it is further assumed that start of a burst occurs with equal probability in all bit positions then the probability of falsely declaring loss of synchronization in this case is approximately 7×10^{-9}. Therefore, the FMU effectively maintains synchronization during a burst.

9.2.3.8 Effect of Noise Environment

The master frame synchronization process described is not necessarily an optimal solution to the problem, but does provide excellent results for the postulated noise environment. Should the FMU be required to operate in an

environment which is significantly noisier (either with regard to the random bit error rate or the extent of bursts), the frame maintenance parameters τ, t and n would need to be reevaluated to reflect this change. As a final point, it should be noted that if a new set of assumptions requires that n be significantly increased, it would probably be advantageous to re-examine the n frame window approach to implementating the maintenance test in an effort to minimize the time to detect loss of synchronization.

9.3 LOOP SYNCHRONIZATION

9.3.1 Problem

The purpose of this section is to examine the performance of digital loop signalling (described in Section 4.1) in the SENET-DAX environment.

9.3.2 Objectives

a. To provide for the transmission of information, supervisory and control signals between subscribers and their serving DAX's

b. To maintain network transparency

c. To examine the requirements of the digital receiver and scanner.

9.3.3 Analysis and Results

9.3.3.1 Discussion

As described in other sections of this report, it appears that future digital telephones will employ cyclically permutable codewords to perform the signalling functions presently performed by analog means (e.g., tones or levels). The attractiveness of these codewords derives from the fact that they are self-synchronizing and that any two codewords ($\geq$ 8 bits) of even parity have a minimum Hamming distance of two. The self-synchronizing capability of these codewords is due to the fact that any cyclic shift of the codeword is still recognizable as the codeword. For example, the three codewords '11000000', '01100000' and '00011000' would be recognized by the switch as the same basic word. However, the price paid for this self-synchronizing feature is a reduction in the size of the code-

word dictionary. Instead of $2^8 = 256$ codewords, the 8-bit premutable dictionary has only 36 words (20 even parity and 16 odd parity).

In general, the local loop signaling plan would be tailored to the noise environment specified. Since no such specification is yet available, the following subsections will demonstrate the performance of digital loop signaling for typical parameters. As will be seen, the performance is more than adequate and there is always the added capability of changing parameters in order to meet future requirements.

9.3.3.2 Supervisory Signal Detection

All Class I channels will be scanned periodically by the digital scanner regardless of the state of the call. It is assumed that the digital scanner uses a 9 out of 16 majority vote by codeword as the criterion for successful supervisory signal detection. Thus the probability of a correct signal detection is given by

$$P_c = \sum_{k=9}^{16} \binom{16}{k} (1 - BER)^{8k} (1 - (1 - BER^{8})^{16-k}$$

where BER is the random bit error rate in the local loop and where it is assumed that each class I channel is scanned every 200 msec for a dwell time sufficient to receive 16 codewords. Note that framing is not necessary since the codewords are self-synchronizing.

Evaluation of P_c for a BER =.05 results in a probability of correct signal detection of 0.86. Although this is not very good performance, it must be remembered that this is for a single scan. The probability that five successive scans result in a correct detection is $1- (1-P_c)^5 = .99996$. Therefore, 99.996 percent of the calls would receive dial tone or release within 1 second (5 x 200 msec), Subject to availability of a digital receiver.

9.3.3.3 False Release

Since each channel is scanned periodically, there is a finite probability that a call will be falsely terminated due to the false detection of a release signal. If it is assumed that the call is end-to-end encrypted then the probability of interpreting the scanned data as a release signal is

$$P_{frs} = \sum_{k=9}^{16} \binom{16}{k} (.5)^{8k} (1 - .5^{8})^{16-k}$$

and the total probability of a false release is

$$P_{fr} = 1 - (1 - P_{frs})^{n}$$

where n is the number of times the average call is scanned during the course of the call. For the assumed five scans per second and for an average hold time of 5 minutes, the probability of a false release is 4.38×10^{-15}.

9.3.3.4 Control Signal Detection

After the digital scanner detects a request for service, dial tone would be returned to the subscriber and a digital receiver connected to the loop to detect the called number. The probability of correctly detecting the called number is a complicated function of the loop noise environment, the numbering plan, and the detection criteria employed. It has been shown [66] that for a 10-digit numbering plan, it is possible to obtain probabilities of correct address detection ranging from approximately 1 to 0.999993 for random bit error rates of .01 to .05, respectively.

9.3.3.5 Information Signals

Information signals (audible tones) provide the subscriber with information relating to the status of his call. It is assumed that such signals would be transmitted from the DAX in the form of digitized tones.

SECTION 10

PERFORMANCE ANALYSIS

SECTION 10

PERFORMANCE ANALYSIS

10.1 TRANSMISSION INTERFACES AND FLEXIBILITY

10.1.1 Problem

The SENET-DAX concept is intended to serve a highly dynamic and evolutionary communication network. In order to perform its mission of an integrated voice and data switching capability, the system must provide a high degree of interoperability with a wide variety of existing systems and equipments employing dissimilar transmission techniques and rates. The flexibility of the SENET-DAX approach in bit orientation and constant frame allocation lends itself to the solution of this problem.

10.1.2 Objectives

The objectives of this portion of the analysis are: to determine interface requirements with other typical communication systems of widely differing transmission technqiues and rates; to identify the salient characteristics of these interfaces; and to identify insofar as possible the methods and constructs that allow interoperability in a modular and efficient manner.

10.1.3 Analysis

10.1.3.1 Loop/Trunk Conversion

The SENET-DAX concept is based on utilization of a constant period, self-synchronizing master frame throughout the DAX network. We have assumed this constant period to be 10 milliseconds, and the trunk transmission bandwidth to be that of T1 Carrier at 15440 bits per frame. The organization of the master frame based on its self-synchronizing feature allows the transmission (on the T1 trunk) of a variety of transmission rates on a non-interfering basis. The maximum individual channel rate possible is primarily limited by the transmission medium bandwidth, since the overhead per frame is a small percentage of the total frame. Thus, a single channel of approximately 1.5 megabits per second (the remaining 440 bits representing overhead), or a multitude of different channels of lower rate, can be

accommodated on the same trunk with the proviso that the total number of bits per frame (including start-of-frame marker, CCIS messages, etc.) does not exceed 15,440.

At each access node the traffic directed at the network is connected to a trunk with appropriate allocation in the 10 millisecond master frame. Table 10-1 presents representative traffic at different rates and the allocation of bits per frame for each rate. At the terminating exchange, each individual channel bit allocation is extracted frame by frame from the trunk, converted into a bit stream, and transmitted to the loop. This loop-to-trunk and trunk-to-loop conversion allows the DAX network to utilize a unique format for internode communications while at the same time accommodate a wide range of loop rates.

10.1.3.2 Intercommunication with Analog Systems

Interfaces with analog systems require the ability to derive or convert signalling and supervision information, and to provide analog-to-digital and digital-to-analog conversion of voice signals.

In order to assess the requirements for interface capability, the signalling and supervision requirements of a broad spectrum of switching central offices are presented in Table 10-2. In this table, except for SENET-DAX, ULS, and AN/TTC-39, systems are characterized by analog voice transmission and a variety of signalling and supervision techniques. The digital interface with AN/TTC-39 and ULS is compatible from the point of view of signalling and supervision as well as the compatibility of digital technique (CVSD 16 or 32 Kb/sec).

A typical architecture for an analog interface has the following characteristics (see Figure 10-1):

a. Variation in signalling schemes is software implemented

b. A hardware adapter allows for 2 to 4 wire conversion and AC tone detection

c. A microprocessor (SSU) or equivalent performs signalling and supervision detection and conversion for a group of lines.

This architecture can potentially allow a universal analog interface concept, with signalling and supervision to be accomplished on a modular basis in an overall distributed architecture.

TABLE 10-1. LOOP/TRUNK CONVERSION

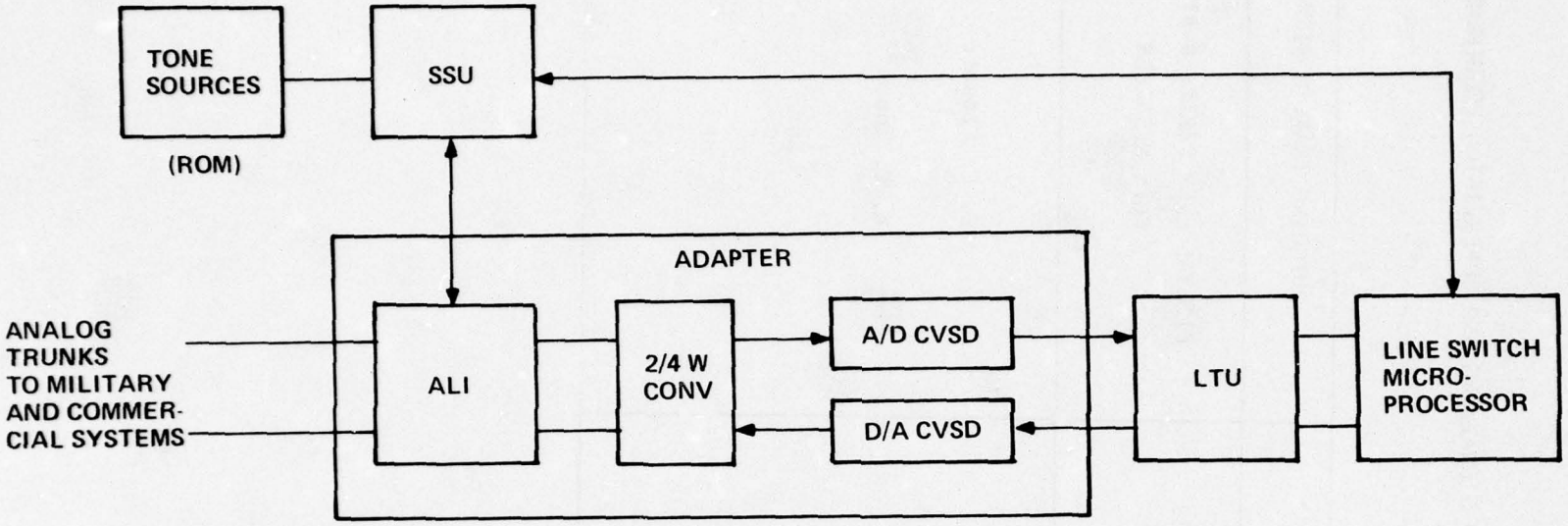
LOOP RATE	TRUNK ALLOCATION BITS PER 10-MILLISECONDS OF T_1 BANDWIDTH	REMARKS
2400 bits per sec	24	Vocoder
16K b/sec	160	DSVT (Future)
32K b/sec	320	DSVT (Present)
48K b/sec	480	6-Bit PCM individual channel rate
50K b/sec	500	KY-3 (Secure digital voice instrument)
56K b/sec	560	7-bit PCM (commercial)
64K b/sec	640	8-bit PCM (commercial rate)
200K b/sec	2000	Slow scan video terminal
1.5M b/sec	15000	Unidentified (Maximum possible)

TABLE 10-2a. INTERSWITCH AND EXTRA-SWITCH SIGNALLING AND SUPERVISION

INTERFACE	SIGNALLING	SUPERVISION	COMMENTS
DAX	CCIS	CCIS	Interswitch CCIS Messages in Class II region
DAX to TTC-39 & TTC-39 to DAX	Digital in-band	Digital in-band	Even parity 8-bit Code words
DAX to TTC-39 & Vice versa	CCIS	CCIS	Out of band signalling supervision for a trunk group
DAX to TTC-39 & Vice versa	DTMF	Tone	Also via satellite
DAX to ULS & ULS to DAX	Digital in-band	Digital in-band	8-bit Code words
DAX to ULS & ULS to DAX	CCIS	CCIS	Out of band signalling
DAX to TTC-38 or TTC-25	DTMF 2/8	Tone	Confirmation type of signalling
TTC-38 or TTC-25 to DAX	DTMF 2/8	Tone	
DAX to AN/TTC-22	Ringdown	E&M or DC loop	Attendent extends it locally
AN/TTC to DAX	Dial pulsing	E&M (SF)	
PABX to DAX	Dial pulsing	D.C. Loop	
DAX to PABX	Ringdown	Loop	Similar to analog 2-wire instrument type 500
Local office to DAX & Vice Versa	Dial pulsing or MF 2/6	E&M	2-wire interface
Tandem office to DAX & Vice Versa	D.C. pulsing or MF 2/6	E&M	2-wire or 4-wire
DAX to AUTOVON	DTMF 2/8	SF	Wink Start
AUTOVON to DAX	SF Dial pulsing	SF	
DAX to manual boards	Ringdown	Ringdown	Attendant extension at manual boards
Manual boards to DAX	Ringdown	Ringdown	Attendant extension at DAX
DAX to SB-3614	DTMF	Tone	
SB-3614 to DAX	DTMF	Tone	

TABLE 10-2b. SUMMARY OF SIGNALLING AND SUPERVISION REQUIREMENTS

INCOMING TRAFFIC SIGNALS	OUTGOING TRAFFIC SIGNALS
DIGITAL EVEN PARITY 8-BIT CODES CCIS MESSAGES	DIGITAL EVEN PARITY 8-BIT CODES CCIS MESSAGES
D. C. Closure A. C. Tones E&M Ring Signal SF	D. C. Closure A. C. Tones E&M SF



ALI = ANALOG LINE/TRUNK INTERFACE
SSU = SIGNALING AND SUPERVISION CONTROL UNIT
LTU = UNIV. LINE TERMINATION UNIT
CONV = 2-TO 4-WIRE CONVERTER

5029-76E

Figure 10-1. Interface with Military and Commercial Analog Trunks

Although a DAX switch is not expected to have analog subscribers, the capability to interface with analog systems can be utilized to allow termination of an analog telephone such as the TA-341, TA-312, TA-236, or commercial type 500 instruments. Therefore, intercommunication between any of these instruments to a digital instrument DSVT is possible (this intercommunication is of course on a non-secure basis).

10.1.3.3 Digital Subscriber Terminal

Voice subscribers to DAX will undoubtedly employ the Digital Secure Voice Terminal (DSVT), and data adapters for this instrument, for circuit switched services. The DSVT is a 4-wire full-duplex local or common battery, 16 pushbutton secure voice instrument. The instrument employs continuous variable slope delta modulation (CVSD) technique to convert analog voice to 32 Kb/s (16 Kb/s) digital baseband bit stream. The digital bit stream is converted to "conditioned diphase" by the modulation process for purposes of transmission. The instrument employs TRI-TAC digital loop signaling and supervision. This technique is based on 8-bit even parity coded sequences. The use of even parity and assignment of one meaning to all unique permutations of a given 8-bit code makes the coding scheme self-synchronizing, such that signal detection and framing are accomplished in the same step.

Intercommunication between subscribers using 16 Kb/sec and 32 Kb/sec CVSD can be accomplished by using converters as a part of shared common equipment to be switched in when needed (based on line classmark). The converter would employ digital techniques for direct conversion betweem 16 Kb/sec and 32 Kb/sec.

On line to trunk calls, the originating DAX establishes a virtual connection and sends forward 160 bits per 10 millisecond frame (the lower rate using less of the frame). The rate conversion between 16 Kb/sec and 32 Kb/sec is accomplished at the DAX interfacing with the 32 Kb/sec subscriber.

10.1.3.4 Digital/Analog Terminals - KY-3

Interoperation with a KY-3 subscriber is as follows. The KY-3 is a 50 Kb/sec PCM full-duplex four-wire encrypted terminal. Signalling and supervision are performed in the clear using 2600 SF signals with on-hook (idle) being represented as presence of 2600-Hz signal and off-hook its absence. Dial pulses are transmitted as

interrupted bursts of 2600-Hz signal at normal rotary dial rates. The ring signal to alert a KY-3 of an incoming call is 1000-Hz.

The DAX interfaces with a KY-3 terminated at a subscriber terminal or via a broadband circuit switch. The signalling and supervision information is detected and converted to DAX format by a microprocessor or equivalent circuit. If the call is directed to a distant switch, the originating exchange establishes a virtual connection and sends forward 500 bits per 10 millisecond frame to the terminating exchange. The voice connection is established on a secure basis, and other than the number of bits per frame transmitted from exchange to exchange, the interlink connection is no different than other connections.

The terminating exchange sends forward a 1000-Hz signal to alert the called KY-3, with signalling and supervision detected and converted at the interface line/trunk circuit. The call proceeds through the DAX network in the same manner as that for two DSVT subscribers at two different exchanges.

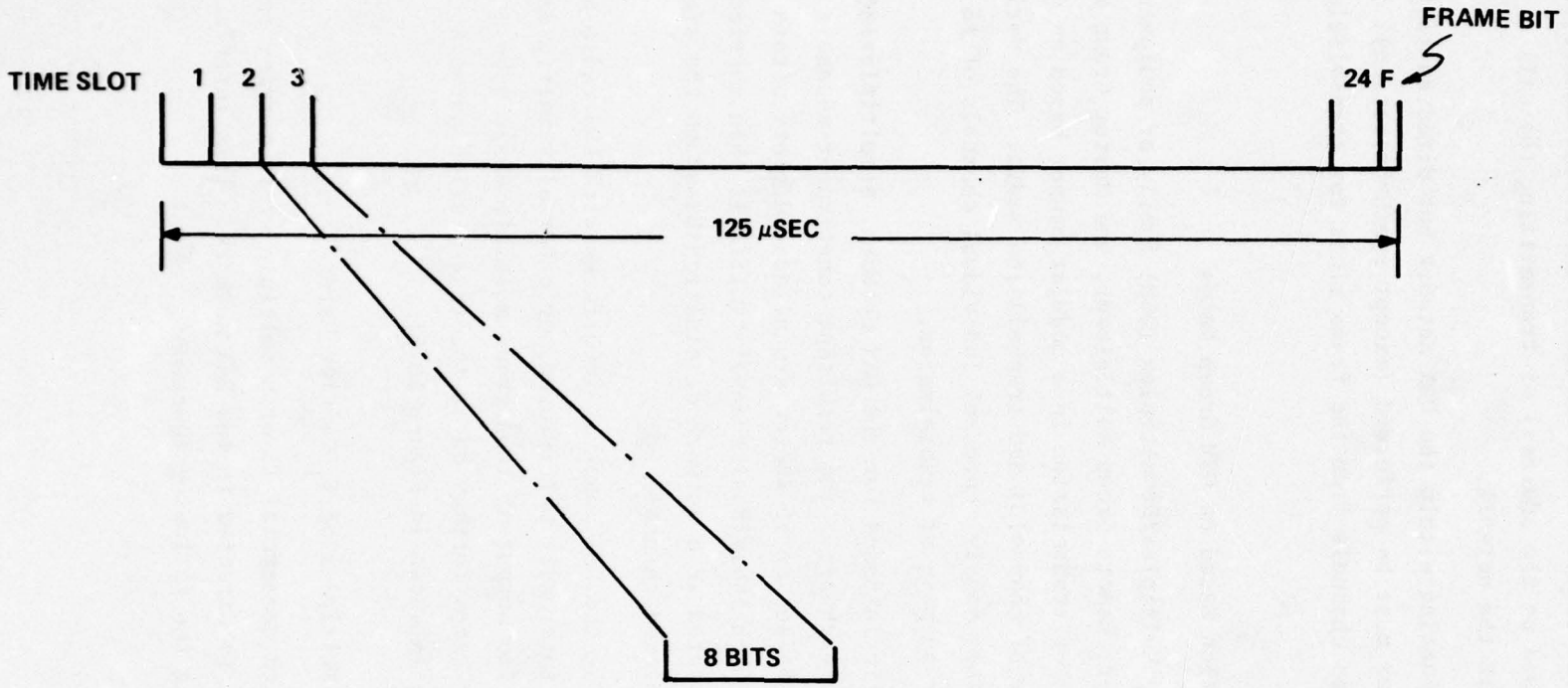
10.1.3.5 Intercommunication With PCM Multiplex

In Pulse Code Modulation (PCM), speech is sampled at an 8 KHz sampling rate. Each sample is represented by a 6, 7 or 8 bit binary code assigning it a discrete amplitude level (one of 256 in 8 bit PCM). The encoding process normally shares a common encoder for a group of channels. Speech samples are first multiplexed into a quasi-digital bit stream, then encoded into a byte-oriented digital format (see Figure 10-2).

Commercial telephone systwms in the United States employ 7 or 8 bit PCM (channel rates 56 Kb/sec and 64 Kb/sec), while certain military systems (TD-660) employ 6 bit PCM (48 Kb/sec). The principles of the DAX interfaces are applicable to all PCM systems regardless of the channel rate.

In order that the traffic in the PCM group multiplex be switched to various points in the DAX network, the following processes must be performed on the incoming multiplex group at the originating DAX:

a. Establish synchronization
b. Demultiplex the group into its constituent channels
c. Derive signalling and supervision information
d. Establish each frame slot at the appropriate number of bits (640,560,480) per 10 msec for transmission.



$$\text{T1 TRANSMISSION RATE} = \{(8 \times 24) + 1\}\, 8000 = 1.544 \text{ Mb/SEC}$$

5028-76E

Figure 10-2. Frame Format for 24-Channel 8-Bit PCM Multiplex

The DAX handles the converted bit stream in the normal manner, either switching it locally (based on the address) or transmitting the call in its own format to any other DAX in the network.

For traffic originating within the DAX network but directed to the commercial network, inverse functions must be performed (except synchronization), in that the gateway switch multiplexes channels from the frame slots for transmission to the commercial network.

10.1.3.6 Intercommunication Based on DGM Group Rates

The Digital Group Multiplex/Demultiplex (DGM) family of equipment such as the Loop Group Multiplexer, Remote Group Multiplexer, and Master Group Multiplexer, provides for optimization of transmission in a modular manner based on distance, amount of traffic (number of channels) and transmission media. The various group and supergroup rates in this family represent individual channels of 16 Kb/sec or 32 Kb/sec multiplexed for purpose of transmission.

On incoming traffic intended for the DAX network, demultiplexing is performed at the originating exchange. The resultant constituent channels having transmission rates of 32 Kb/sec or 16 Kb/sec are allocated slots within the 10 millisecond time frame. Each individual channel traffic is then carried as subscriber traffic and terminated at a network destination based on the signalling derived from the demultiplexed channel.

For traffic leaving the DAX, such as interfaces with LOS radio and satellites, the loop channels can be internally multiplexed, or externally multiplexed using a loop group multiplexer. The output of loop group multiplexers can be multiplexed into group and supergroup rates further on in the transmission network. This concept of the DGM interface is shown in Figure 10-3.

10.1.3.7 Interface With Multiples of T_1 Carrier Rate

In interfacing with commercial T1 or a multiple of T1 carrier rates, it is assumed that this traffic is directed to the DAX network. This interface is accomplished by performing the following functions:

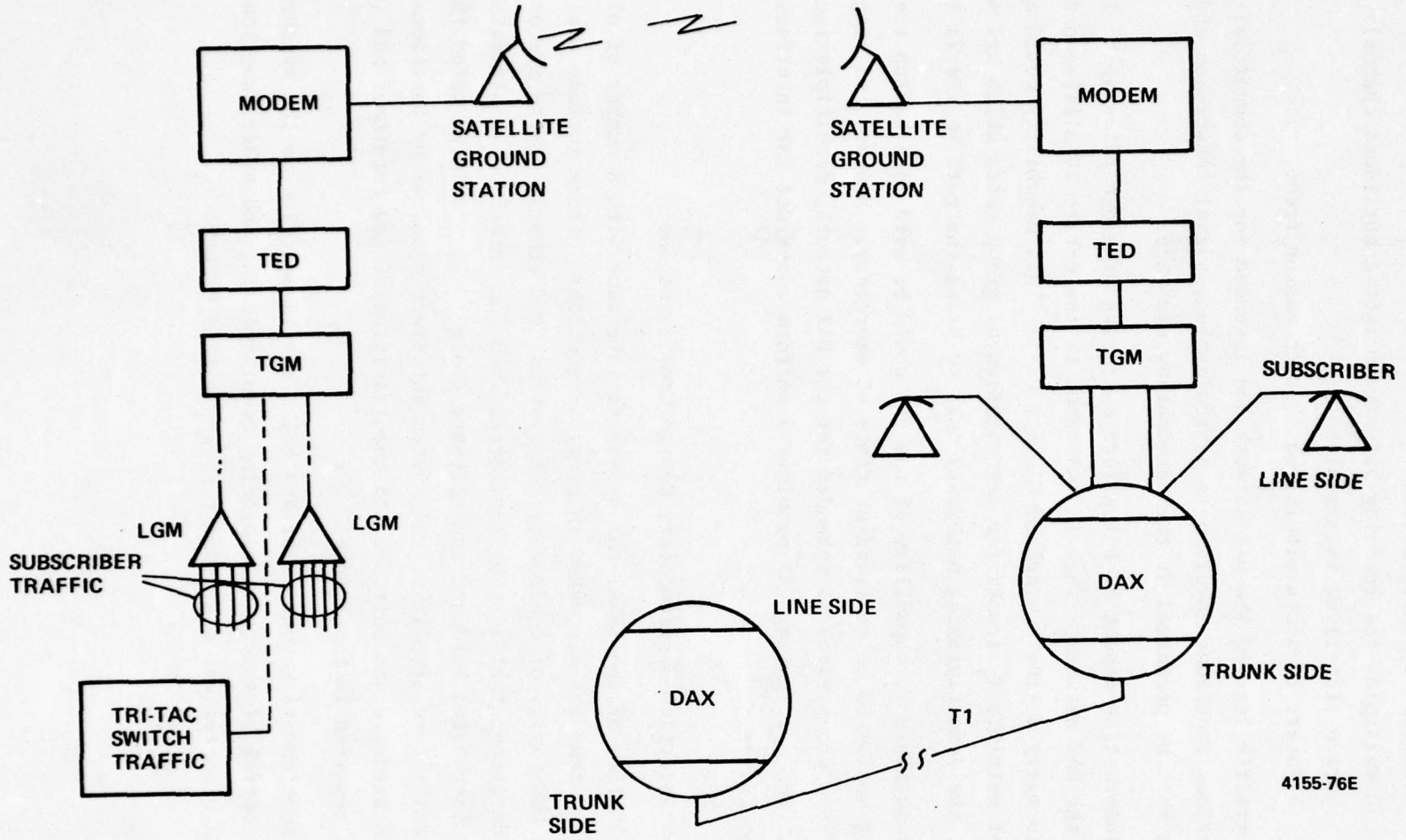


Figure 10-3. DAX DGM Interface

a. Establish synchronization

b. Demultiplex the incoming bit stream into constituent channels

c. Derive signalling information

d. Allocate to slots within the 10-msec master frame.

For traffic leaving the DAX network and intended for the commercial network, the various functions involving synchronization, signal conversion and multiplexing must be performed at the terminating exchange.

An interesting aspect of DAX allows switching of group rates up to 1.5 Mb/sec through the DAX network. This is analogous to the entire 10-millisecond frame being used to carry a single wideband Class I Call. This capability could allow assignment of multiple T_1 trunks for transmission of group rates which are multiples of T_1 rates, the demultiplexing necessary only to bring the rate to the T1 level.

This multiple T_1 capability of the DAX could be used as a backup transmission and switching medium to be employed at times of emergency. However, as long as the traffic in these group rates is intended for the DAX network, demultiplexing and conversion at the originating DAX provides a uniform construct for interfacing with all group rates.

10.1.3.8 DAX Interface Supplementary Design Considerations

The ability of the SENET-DAX system to interface with a number of different transmission systems was discussed in previous sections. These include analog systems, the DGM family of equipment, PCM systems, and others. There are other design considerations that must be considered when the interface is ultimately realized in distributed hardware and software design. It should be noted that SENET-DAX architecture simplifies the system interface problems by functionally separating in hardware and software the peculiarities of new terminals and interfaces which may be required in the future.

The supplementary considerations below are in addition to the synchronization, error control, framing, multiplexing/demultiplexing and other functions of the SENET-DAX architecture discussed throughout this report.

10.1.3.8.1 Signaling and Supervision Conversion - The function of signaling and supervision conversion is clearly important for interoperability of a variety of systems in a highly evolutionary environment of communication networks. While the DAX constant period frame provides for inter-DAX communication, with its unique synchronizing and SOF marker format, the DAX must also be provided the ability to communicate with existing systems and equipments in the DCS. Therefore, a DAX interface must have the ability to perform and convert a variety of signaling and supervision functions characteristic of such systems and equipments. Interactive hardware and software will be required for execution of this function.

10.1.3.8.2 Line and Signal Conditioning - Modulation and demodulation functions (conversion of digital baseband signal to conditioned diphase and vice versa) for interfacing with the transmission media, and such functions as amplification and 2- to 4-wire conversion, for interface with analog systems, are ordinarily performed by the interface circuit.

10.1.3.8.3 Line Protection Function - The interface must contain circuit hardware for protection of the system against voltage and current surges from lightning and other external sources.

10.1.3.8.4 Isolation, Patching, Monitoring Function - These functions are necessary for test and maintenance of a communication system.

10.2 TRANSMISSION OVERHEAD

10.2.1 Problem

The prime function of a switching system is to decrease unit communication costs by increasing the utilization of transmission facilities and thereby decreasing transmission costs for a given traffic load, quality and type of services. It is the purpose of this section to analyze the sources of transmission overhead which limit the transmission efficiency of typical FTDM trunks connecting nodes in a DAX network.

10.2.2 Objectives

a. To properly define transmission efficiency and overhead for a FTDM DAX trunk which carries a mix of voice and data traffic in circuit, packet and message switched modes, each having different accuracy, response time and other performance requirements.

b. To determine quantitative relationships relating overhead and efficiency to DAX parameters, noise and traffic environment, and to other elements of DAX design and operational concept.

c. To optimize efficiency for packet switched data with reference to packet size.

d. To calculate transmission efficiency for illustrative examples of Class I and Class II traffic in a postulated baseline system thereby providing guidance for the recommendation of DAX design concept.

10.2.3 Analysis and Results

10.2.3.1 Definitions

10.2.3.1.1 Transmission Overhead of a DAX trunk

A DAX trunk is a full duplex secure multi-channel digital link between DAX switches using the SENET concept. Traffic carried consists of virtual circuit switched secure voice, video and fax (Class I) and data traffic which is packet switched message traffic (Class II). Class I traffic is typified by the fact that it must be sent in a time transparent mode through the entire holding time of the call. If the output data rate of the Class I terminal is R bits per second and F is the frame interval of the DAX trunk, then each frame on every trunk over which the call has been routed must contain $F \cdot R$ bits in each direction from the instant the virtual connection is cut through until it is broken down. From the viewpoint of transmission efficiency all data transmitted in

both directions in Class I slots will be considered essential traffic even if it consists of idle periods, forward error control (FEC), or even noise. In other words the essential Class I traffic is considered to be all the bits transmitted in the allocated bidirectional slots regardless of content or transmission errors. The Transmission overhead relating to Class I traffic is defined as the information that has to be transferred across the trunk to establish circuits and dynamically control the position of any channel within the frame in order to permit compacting of the Class I region in the frame as virtual connections are made or broken down in the network. This overhead traffic is carried by CCIS messages which are treated as a special type of Class II packet switched traffic. Other types of overhead information which have to be transmitted in the master frame are frame marker sequences and other control sequences which may be used to indicate changes in the class of traffic occurring during a frame. Administrative, maintenance and test traffic required to monitor, control, and adjust network or trunk operation would also be transmission overhead that would be carried as another source of service traffic on the Class II data section of the DAX.

10.2.3.1.2 Bit Stuffing

It may be beneficial to restrict the slot sizes to integral multiples of some convenient word size or to accommodate a terminal having an output data R such that R·F is not equal to an integral number of bits or word size. Bit stuffing consists of adding a sufficient number of bits to the output stream to change R to a higher value R' which will produce an integral number of bits or words per frame. The stuffed bits are identified and removed by the output terminal and the output reclocked to retain time transparency. The stuffed bits which are carried within Class I channels in a frame are to be considered as transmission overhead rather than essential traffic.

10.2.3.1.3 Class II Data Traffic

One of the characteristics of class II traffic is that it is of use to the user only if it is error free within acceptable confidence levels in the expected bit and burst error environment of the trunk. Hence in the case of all Class II traffic (including CCIS and other service traffic) the essential traffic rate (R_e) is defined as the maximum rate at which correct data can be throughput to the user over the trunk. Transmission overhead rate (R_o), in the case of Class II

traffic, is defined as the difference between the trunk data transmission rate, as determined by the modem, and the essential user traffic rate R_e. The Class II transmission overhead includes the following: (1) the non-information bits that must be transmitted over the trunk for the purpose of message, packet, network and error control and synchronization; (2) retransmission of packets delivered in error; (3) acknowledgment and call initiate packets; (4) the time the trunk must remain idle in an ARQ block by block mode waiting for a decision as to the next packet to be transferred. In an ARQ continuous mode there is no waiting time but the transmission overhead must then include the time for retransmission of correct packets.

10.2.3.1.4 Transmission Efficiency (E_{ff})

The transmission efficiency of a trunk is defined as the ratio of essential user traffic (R_e) to the trunk modem rate (R).

$$E_{ff} = \frac{R_e}{R} = \frac{R_e}{R_e + R_o} \tag{1}$$

In view of the difference in definition of R_e for Class I and Class II traffic, both of which are to be simultaneously carried over a DAX trunk, we will define the transmission efficiency of a DAX trunk as the weighted average of the Class I and Class II efficiency. The weighting factors being the percentage of each class of traffic in the total trunk traffic. Thus the DAK trunk efficiency is given by:

$$R_{ff}\ (DAX) = \frac{E_I R_{eI} + E_{II} R_{eII}}{(E_I + E_{II})\ R} \tag{2}$$

where E_I and E_{II} are the equivalent Erlangs of class I and class II traffic and R_{eI} and R_{eII} are the essential traffic rates for classes I and II.

The DAX transmission overhead is likewise defined and is hence given by:

$$R_o\ (DAX) = \frac{E_I\ R_{OI} + E_{II} R_{OII}}{E_I + E_{II}} \tag{3}$$

Equation (2) can be used to compute the trunk transmission efficiency for given values of E_I and E_{II} traffic loads.

10.2.3.2 Optimization of Transmission Efficiency

10.2.3.2.1 Designation of Parameters

Parameters which contribute to transmission overhead in the concept of the dynamically allocatable frame will now be identified.

10.2.3.2.1.1 Transmission

Characteristics of various transmission paths and the nature of the error control and correction procedures used will have an effect on transmission efficiency since errors in transmission will be considered here as overhead. The effect of errors depends on the noise and delay characteristics of the transmission path employed. Some media are relatively error free, e.g., line-of-sight, while others are quite noisy, e.g., troposcatter. Some exhibit long delay, such as satellite relays. Various transmission environments which the DAX will be required to utilize are as follows: wireline, line-of-sight, undersea cable, satellite (including TDMA) relay, troposcatter, and microwave link. It is not possible to prevent all the errors occurring in the data transmitted across a particular channel. However, in most cases these errors can be detected and powerful correction procedures applied to correct them.

Criteria for Class I transmission, such as voice, is considerably different from those for data. The voice signal can undergo a considerable degree of noise and distortion without any appreciable effect on the intelligibility of the speech. The robustness of voice is such that it degrades gracefully and does not become annoying, or even too unnatural to listen to, as the signal-to-noise ratio decreases to very low values.

Likewise, the facsimile message and video signal contain a certain degree of redundancy similar to voice. The presence of a few incorrect symbols due to noise will still allow the message to be readable and understandable.* Some Class II traffic exhibits this characteristic (e.g., Teletype); however, generally Class II traffic must be error protected.

---

* If more protection is required, FEC systems can be employed.

Packet error rates vary as a function of message length among other factors. When error-detection is used with Class II data retransmission, the number of packets or blocks that will have to be retransmitted is a function of the message length. If very short messages are sent, retransmission will not be required for most of them. But if the messages are long enough, a high proportion will be in error and have to be retransmitted. In this instance it is an advantage that errors tend to cluster since this tends to increase the proportion of error-free messages.

There is a timing advantage to be gained, however, in transmitting large blocks. The number of non-information bits per packet tends to remain constant as does the length of waiting time to receive and ACK or NAK from the receiving node. Therefore the larger the block of information in a packet the greater will be the transmission efficiency in an error free environment. The specific cost in terms of transmission efficiency of error protection is dependent upon the type of ARQ used as well as the packet size and bit error rate of the medium. In block by block ARQ, the waiting period is lost even when there is no errors. In continuous ARQ with complete retransmission after a NAK the waiting period is lost only when a packet error occurs and in continuous ARQ with retransmission of only the errored packet there is no channel outage for waiting for an ACK/NAK. An optimum packet size exists for each type of ARQ and a given bit error rate on the trunk.

10.2.3.2.1.2 Error Bursts

A considerable portion of bit errors are clustered in bursts rather than being uniformly distributed. Sometimes these bursts last for hundreds of bits. Burst data may be defined by a burst error rate and a duty cycle, or the percentage of time that the burst condition exists. The effect of these errors are modeled as follows: For long periods of time there exist relatively normal error rate transmission where the probability of an error is of the order of 10^{-3} or less. During this time a large proportion of messages may have no errors occurring in them at all. Clustering of errors occur with a burst duty cycle of 5% during which time the error rate is very high, (e.g., 0.2), effectively making all the data within the burst unusable. It is possible for a noise burst to affect more data than appeared during the burst. For example, any errors in a packet will result in that packet being entirely rejected and resent via ARQ effectively causing all the data comprising the packet to be considered overhead. Occasionally, an error burst could be situated such that its duration straddles two packets, rendering both useless and thereby reducing the overall efficiency of the system accordingly.

10.2.3.2.1.3 Error Correction

It is generally found that reliable communication over data channels cannot be achieved without some form of error correction. Two types of error control and correction procedures are typically employed to improve the reliability of transmitted data - Forward Error Correction (FEC) and/or Automatic Repeat Request (ARQ). FEC is a method which attempts to determine the location of the errors from the pattern of discrepancies using structural redundancy encoded into the message. Systems employing FEC then can correct these errors, up to some maximum number, depending upon the strength of the error protection code used. ARQ systems use either parity or polynomial check sequences to detect errors and, when an error is detected within a packet, retransmission of that packet is requested. If no discrepancies are found, the packet is delivered to its destination and the DAX which receives it notifies the transmitting DAX of its acceptance via an Acknowledge (ACK) message over the return path. If discrepancies exist Negative Acknowledge (NACK) messages are transmitted. This causes retransmission of the packet until an ACK message is received. Under this scheme erroneous data is delivered to the destination only if the decoder fails to detect the presence of errors.

Generally ARQ systems provide more reliable error protection than FEC systems[16]. ARQ schemes are reliable and relatively insensitive to conditions on the channel. However, while very powerful and effective against moderate error and short delay conditions, ARQ may cause intolerable loss of transmission efficiency under high noise and/or long path delay conditions in half duplex links or when storage limitations prevent the use of continuous mode of operation.

There are three types of ARQ detection retransmission schemes * - Stop and Wait ARQ and two types of Continuous ARQ. In Stop and Wait ARQ, after sending a packet, the sending DAX waits for an Acknowledgment (ACK) from the receiving DAX before sending another packet, or in the case of receipt of a NACK, before retransmitting the same packet again. This situation is illustrated in Figure 10-4. Although

* Autodin uses a semi-continuous mode of ARQ where transmission is allowed to continue but a wait mode is implemented if the ACK is not received before the completion of the next packet.

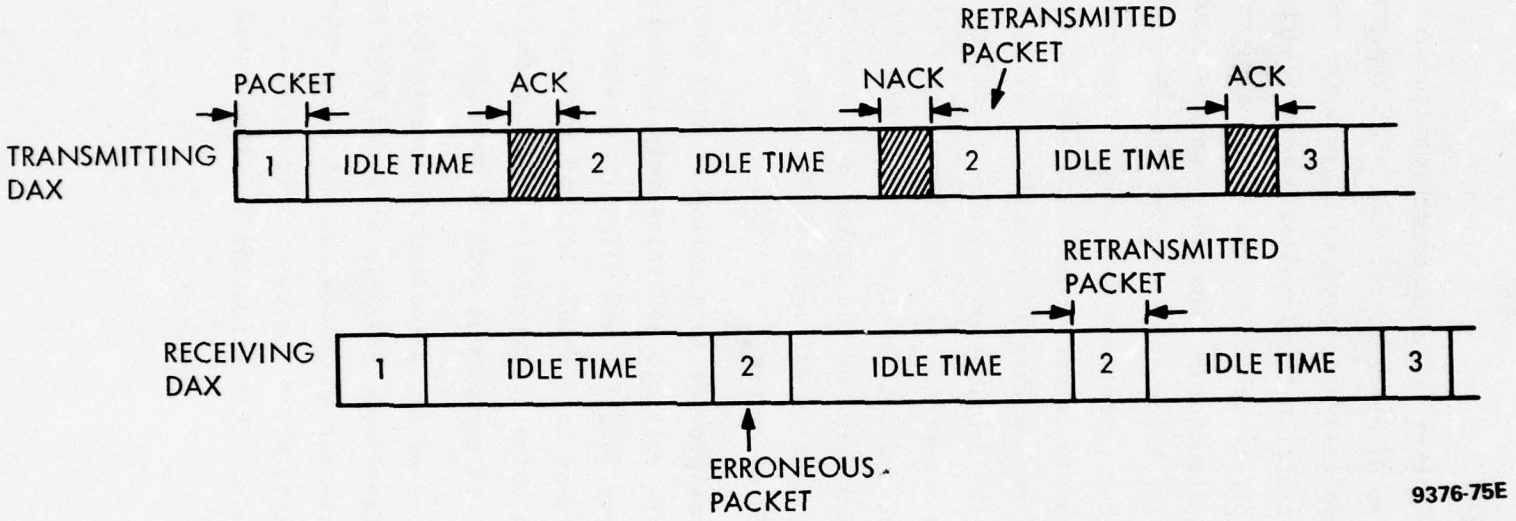


Figure 10-4. Stop and Wait ARQ Retransmission

very simple, this scheme is inefficient due to the idle time spent waiting for acknowledgements from transmitted packets. Under high noise (Troposcatter) or long path delay conditions (satellite), this inefficiency could turn out to be unacceptable.

In the continuous type of ARQ system, the sending DAX need not wait for an acknowledgement after sending a packet. As soon as it finishes transmitting one packet, it sends the next one. When a Negative Acknowledgment (NACK) for a particular packet is received, the sending DAX can pull back and retransmit either the erroneous packet and all packets transmitted in the intervening time between the original transmission of the erroneous packet and the receipt of the NACK (see Figure 10-5a) or the sending DAX can transmit only the erroneous packet in question (see Figure 10-5b)*. Retransmitting only the packet that was detected in error produces more efficient operation. In Figure 10-5, the transmitted and received sequence for continuous ARQ retransmission are shown. Block 2 is shown as received in error. Using the technique shown in Figure 10-5a, the sending DAX retransmits packets 2-5 while Figure 10-5b, only packet 2 is retransmitted. Upon valid receipt the packet is reinserted into the proper place in the message via sequence numbers. These ARQ error control schemes are examined more quentitatively in later sections.

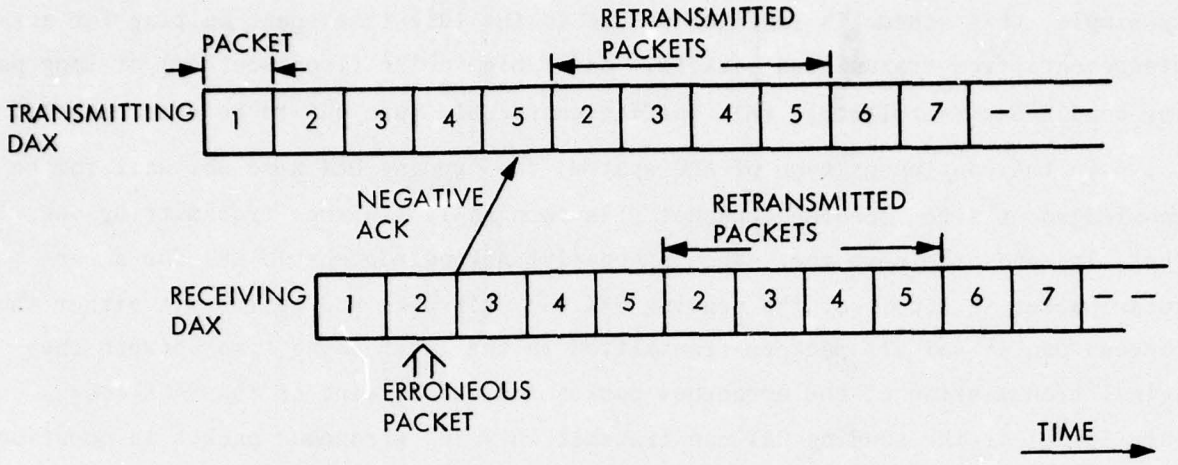
10.2.3.2.1.4 Traffic Parameters

One of the characteristics of bit errors is that as the transmission rate is increased the error rate also becomes greater. At very high speed the error rate increases rapidly and one pays heavily for attempting to extract minimum line efficiency. There exist, however, methods such as advanced high speed modem techniques, which alleviate the problem to some degree.

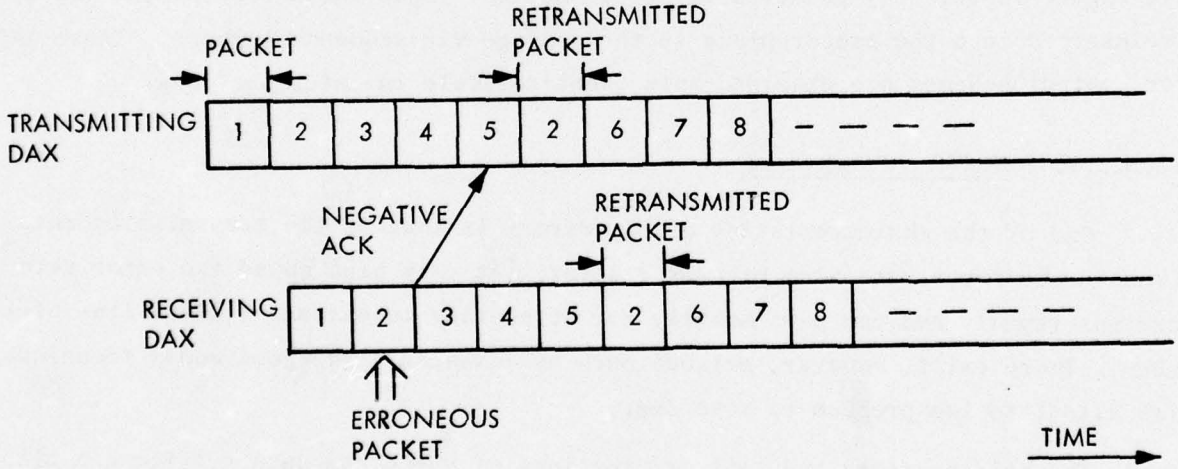
The holding times and call originations of voice and data traffic can also impact on the transmission overhead rate. Increased call rates require more signaling resulting in an increase in the number of bits used for signaling overhead. Under extreme congestion the overhead from originating traffic can become large enough to restrict throughput. Originating calls which are blocked or excessively delayed, are tried again, thereby increasing the signaling load, even though they are not being served.

---

* It is assumed that sufficient buffering is provided at the transmitting DAX to accomplish this.



(A) RETRANSMITTING ALL PACKETS AFTER ERROR



(B) RETRANSMITTING ONLY THE PACKET IN ERROR

9377-75E

Figure 10-5. Continuous ARQ Retransmission

10.2.3.2.1.5 CCIS Messages

Overhead for Class I traffic includes the Class II data load represented by Class I CCIS messages. This includes those messages (packets) used in the establishment and termination of calls, coordination and control of the frame, special messages to compact the allocation of bits in the frame after a call termination, propagation of trunk signalling messages, etc. CCIS messages which are used to establish and terminate Class II calls are also considered to be overhead.

Other messages which utilize the CCIS subregion and which are considered to be overhead include the following:

- Achievement of call synchronization
- Resynchronization after an out-of-sync condition
- Crypto synchronization and resynchronization
- Network control (SYSCON & TECHCON)
- Maintenance messages
- Transient and catastrophic recovery messages (initialization, startup, etc)
- Routing messages
- Accountability messages: memory map verification; AMA for Class I (called party, calling party, duration of call, precedence, etc); AMA for Class II (parties, length of message, precedence, etc); journal entries for archival storage
- Error control procedures; ACK, NACK (retransmit ARQ), etc.

10.2.3.2.2 Optimization of Transmission Efficiency of Class I

10.2.3.2.2.1 Bit Stuffing

In Section 10.2.3.1.2 we defined as overhead the stuffed bits carried by Class I channels when these channels are specified in minimum increments, called the slot size. Here we will examine quantitatively the relationship between slot size and choice of frame interval and their effect on transmission overhead.

In Table 10-3, the number of bits required for a given master frame interval have been tabulated for an assumed distribution of Class I terminal rates. For example, a 2400 bps terminal transmission rate requires a 24 bit channel for a 10 msec frame interval, i.e.,

TABLE 10-3.

NUMBER OF BITS PER FRAME INTERVAL FOR A GIVEN DISTRIBUTION OF VOICE TERMINALS

Percent Distribution	Terminal Transmission Rate	Frame Period (ms)								
		1	5	8	9	10	11	12	15	20
10%	2400 b/s (Vocoder)	2.4 b/f	12 b/f	19.2 b/f	21.6 b/f	24 b/f	26.4 b/f	28.8 b/f	36 b/f	48 b/f
10%	4000 (LPC)	4	20	32	36	40	44	48	60	80
15%	8000 (APC)	8	40	64	72	80	88	96	120	160
50%	16000 (CVSD)	16	80	128	144	160	176	192	240	320
10%	32000 (CVSD)	32	160	256	288	320	352	384	380	640
5%	50000 (KY-3)	50	250	400	450	500	550	600	750	1000

$$\frac{\text{frames}}{\text{sec}} = \frac{1}{10 \times 10^{-3}} = 100$$

$$\frac{\text{bits}}{\text{frames}} = \frac{2400 \text{ b/s}}{100 \text{ f/s}} = 24 \text{ bits/frame}$$

The distribution of Class I terminals is estimated, based on DOD projected data for 1985. Note that fully half of the digitized voice terminals at that time are projected to be 16 kbps Continuous Voice Delta Modulation (CVSD). Others include terminals using 32 kbps CVSD, Linear Predictive Coding (LPC), Adaptive Predictive Coding (APC), 2400 bps Vocoder techniques, and KY-3 secure voice Pulse Code Modulation (PCM). Facsimile and video terminals were not considered.

The transmission overhead inefficiency due to bit stuffing will now be calculated. Consider a frame interval of 5 msec and let us examine the case for a slot size of 16 bits. From Table 10-3, the 2400 bps Vocoder requires 12 bits per channel for a 5 msec frame interval. Since bit durations are specified in minimum increments of 16 bits, 4 bits for each channel are wasted and must be stuffed for transmission purposes. At the receiving end the reverse of the stuffing process takes place. No information is transmitted by the stuffed bits; they are considered pure overhead. Likewise, for the 50 kbps KY-3 secure voice terminal, 6 bits per channel must be stuffed. The inefficiency is calculated as follows:

$$\text{Ineff.} = \sum_{i=1}^{N} \frac{Si}{Bi} \times Di$$

where Si = bits stuffed per frame per channel for terminal type i

Bi = total bits per frame per channel for terminal type i

Di = % distribution of terminal type i

N = Number of terminal types.

For our example

$$\text{Ineff.} = (4/16 \times 0.1) + (12/32 \times 0.1) + (8/48 \times .15) + (0/80 \times .5) + 6/256 \times .05) = .0887$$

Thus the inefficiency due to bit stuffing for a 16 bit slot size and a 5 msec frame interval is 8.87%. Table 10-4 gives the inefficiency for several values of slot size and frame period. Combinations of slot size and frame interval which yield efficient transmission (low overhead) are indicated in the table within the dotted region.

On the basis of the results shown in Table 10-4 it appears that the smaller a slot size chosen, the better the efficiency from a frame utilization point of view. However, there are other considerations that may enter into the choice of slot size. For example, logic or processing hardware may already be standardized on a particular transmission increment, such as 8-bit bytes, rather than on some other increment. Another advantage is the efficiency of a large increment in terms of the overhead bits needed to describe the positioning and allocation of Class I calls in the memory map listing * . This, however, would not effect transmission overhead but would impact processor memory. Finally, slot sizes in the region of low overhead (1, 2, 4, 8, and 16 bits/slot - see Table 10-4) not only provide efficient frame utilization in the Class I region but also do not add overhead bits in the Class II region for Class I & II CCIS messages. This is because CCIS messages (packets), as will be shown in the next section, turn out to be a integral factor of these slot sizes.

10.2.3.2.2.2 CCIS Messages

We will now examine the effect of CCIS messages on transmission overhead.

---

* To identify call

Bits required (B.R.) = $\log_2 \left(R/R_o\right)$

where R = carrier rate, Ro = minimum terminal rate

R/Ro = maximum number of calls per frame

To identify starting position

Starting position (S.P.) = $\left\lfloor \log_2 \frac{(R)\bullet(T)}{W} \right\rfloor$

where R = carrier rate, T = frame period, W = slot size

Example: Let R = 1.544×10^6 bps (T1 carrier rate), Ro=2400bps, T=10 msec, W=8 bits

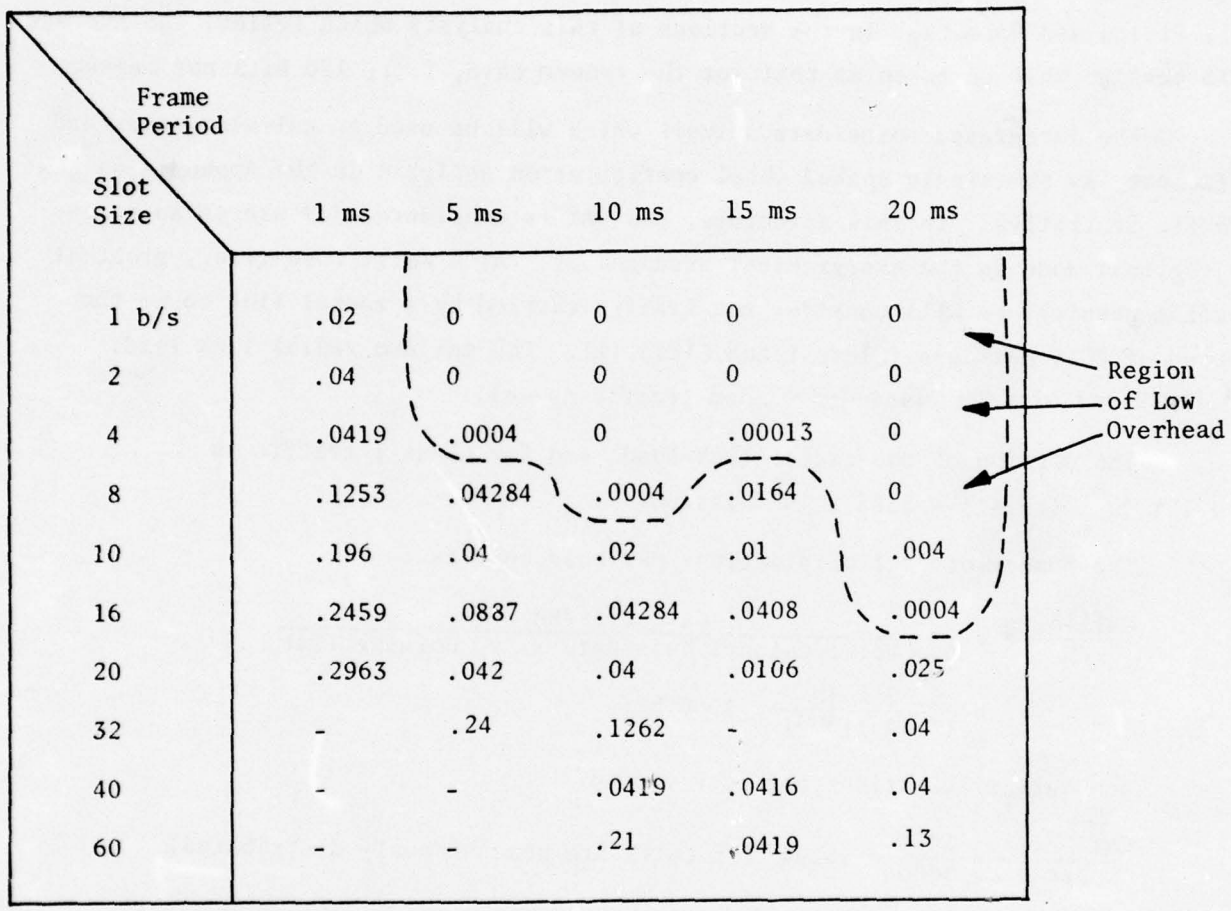
B.R = $\left\lfloor \log_2 (1.544 \times 10^6 / 2.4 \times 10^3) \right\rfloor$ = 10 bits to identify call

S.P = $\left\lfloor \log_2 \frac{(1.544 \times 10^6 \cdot 10 \times 10^{-3})}{8} \right\rfloor$ = 11 bits to specify starting position in the frame

TABLE 10-4.

EFFICIENCY VERSUS SLOT SIZE AND FRAME PERIOD

Slot Size \ Frame Period	1 ms	5 ms	10 ms	15 ms	20 ms
1 b/s	.02	0	0	0	0
2	.04	0	0	0	0
4	.0419	.0004	0	.00013	0
8	.1253	.04284	.0004	.0164	0
10	.196	.04	.02	.01	.004
16	.2459	.0887	.04284	.0408	.0004
20	.2963	.042	.04	.0106	.025
32	-	.24	.1262	-	.04
40	-	-	.0419	.0416	.04
60	-	-	.21	.0419	.13



10.2.3.2.2.2.1 Class I CCIS Messages (Packets)

For every completed Class I call, one of the message sequences in Tables 10-10-6 or 10-7 are transmitted depending on whether the DAX is originating a call, providing tamdem service, or terminating the call. This is depicted graphically in Figure 10-6 where each CCIS message is transmitted as an ADCCP packet. Besides the CCIS messages the number of bits per message are given as determined in Section 2.4 CCIS Fields and Formats. In the sections of this analysis which follow, the average CCIS message will be taken as that for the tandem case, i.e., 150 bits per message.

The integrated voice/data network which will be used to calculate overhead efficiency is the single spoked wheel configuration analyzed in the Appendix on DAX Traffic Statistics. In this structure, the DAX is considered for use as an access or regional node in the hierarchical arrangement. As a worst case (i.e., greatest traffic density) we will consider the traffic carried by a radial link to be the source of CCIS messages (Class I and Class II). The maximum radial link load, 10.1 x 9 and bits/BH, includes tandem traffic as well.

The portion of the radial link load used for Class I traffic is $(10.1 \times 10^9)\ (.8741) = 8.83 \times 10^9$ bits/BH. *

The number of call originations per busy hour is

$$\frac{\text{Call-orig}}{\text{BH}} = \frac{\text{Call-Bits/BH}}{\text{Avg Voice Channel Data Rate x Avg Holding Time}}$$

$$= \frac{8.83 \times 10^9}{(15440)(300)} = 1906 \text{ bits}$$

and call originations per busy second

$$\frac{\text{Call-orig}}{\text{sec}} = \frac{1906}{3600} = .5294 \quad \text{(if calls are homogeneously distributed).}$$

The number of CCIS messages per call origination is 11; therefore the

Messages/Sec = (.5294) (11) = 5.823

and

Bits/sec = (5.823) x 150 bits/CCIS msg = 873.45.

* From Traffic Analysis Appendix

TABLE 10-5.

CCIS MESSAGE SEQUENCE FOR COMPLETING CLASS I CALLS

MESSAGE SEQUENCE OF A TANDEM SWITCH

MESSAGE #	Message	Phase	# of bits/msg
1	Reservation Request to next switch	Forward	224
2	Reservation Agreement to proceeding switch	Reservation	168
3	Reservation Acknowledgement to next switch	Phase I	104
4	Ringback Request to next switch	Ringback	168
5	Ringback Acknowledge to previous switch	Phase II	104
6	Allocation Request to next switch	Allocation	168
7	Allocation Agreement to preceeding switch	Phase III	168
8	Allocation Acknowledge to next switch		104
9 *	Drop Request to next switch	Clearing	168
10	Drop Agreement to preceeding switch	Phase IV	168
11	Drop Acknowledge to next switch		104
			1648
		$\overline{\text{Avg}}$	= 149.81

Average number of bits per CCIS message $\cong$ 150

* Assume that called subscriber has gone on-hook first.

TABLE 10-6.

CCIS MESSAGE SEQUENCE FOR COMPLETING CLASS I CALLS (Cont'd)

MESSAGE SEQUENCE AT ORIGINATING SWITCH

MESSAGE #	Message	Phase	#of bits/msg
1	Reservation Request to next switch	I	232
3	Reservation Acknowledgement to next switch		104
5	Ringback Acknowledge to previous switch	II	104
7	Allocation Agreement to preceding switch	III	168
11(7)	Drop Agreement to preceding switch	IV	168
			776
			Avg = 155.2

TABLE 10-7.

MESSAGE SEQUENCE AT TERMINATING SWITCH

MESSAGE #	Message	Phase	#of bits/msg
2	Reservation AGreement to preceding switch	I	168
4	Ringback Request to next switch	II	168
6	Allocation Request to next switch	III	168
8	Allocation Acknowledge to next switch		104
9	Drop Request to next switch	IV	168
11 *	Drop Acknowledge to next switch		168
			944
			Avg = 157.33

* Assume that called subscriber has gone on-hook first.

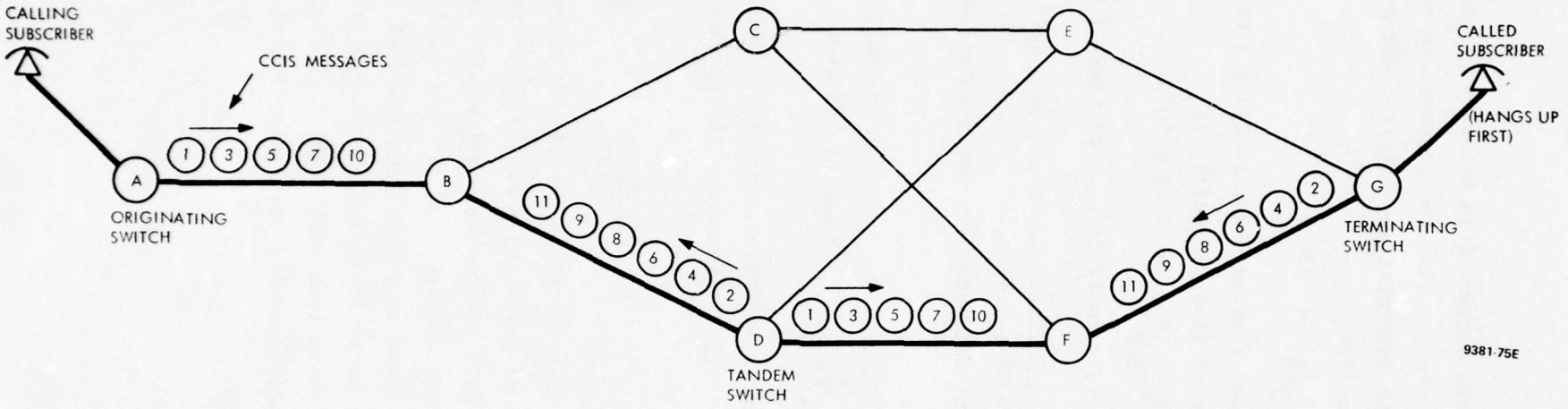


Figure 10-6. CCIS Messages Transmitted During Typical Class I Call Sequence (See Table 10-9 for Definition of Messages)

This number takes into account CCIS messages transmitted over two links; the link to the next switch and the link to the previous switch (see Tandem switch in Figure 10-6). In order to determine the bits/sec for one radial bidirectional link we must divide by two. Therefore

$$\text{Bits/sec} = \frac{873.45}{2} = 436.73.$$

This is the equivalent transmission rate due to Class I CCIS traffic. The equivalent transmission rate of Class I traffic is

$$\text{Bits/sec} = \frac{8.83 \times 10^9}{3600} = 2.45 \times 10^6.$$

The transmission overhead (i.e., inefficiency) of Class I CCIS messages is given by

$$1 - \text{Eff}_I = \frac{436.73}{436.73 + 2.45 \times 10^6} = 1.78 \times 10^{-4} = .000178$$

or approximately .018%. Thus CCIS messages are a relatively small percentage of the Class I traffic.

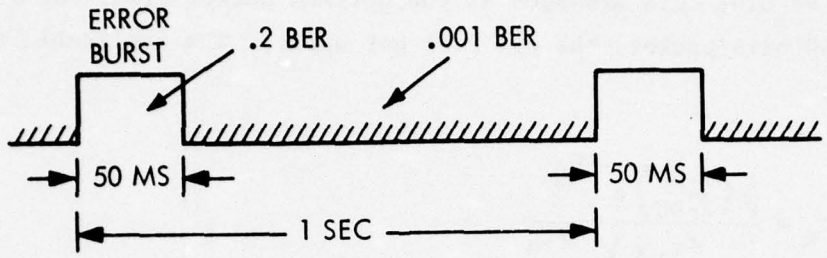
On the average, a CCIS message occurs at the following rate

$$\frac{150 \text{ bits/CCIS msg.}}{436.73 \text{ bits/sec}} = .343 \text{ sec or } 343 \text{ msec},$$

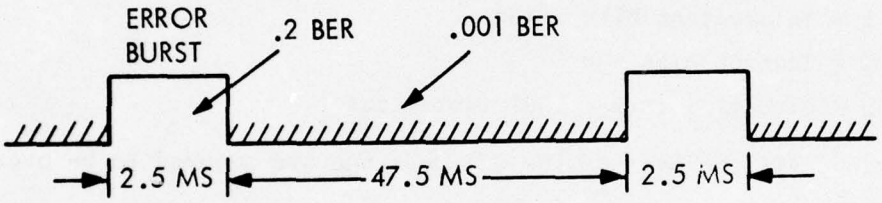
or roughly 1 CCIS message every 34 frames. Since, on the average, we have 5-1/2 CCIS messages per call origination (See Figure 10-7), there is one call origination and termination every 189 frames, i.e.

$$34.3 \frac{\text{frames}}{\text{CCIS msg}} \times 5\text{-}1/2 \frac{\text{msg's}}{\text{call orig}} = 188.6 \frac{\text{frames}}{\text{CCIS msg}}$$

In Figure 10-7 we consider these results in light of the error environment in which we must be capable of operating. Bursts occur from a 1 Hz to 20 Hz rate during which time all data within the burst window is assumed lost. A CCIS message (packet) occurs every 343 msec, and lasts for .01 msec (see Figure 10-7B). When an error burst occurs there is a high likelihood of the message being destroyed completely since the burst lasts for a much longer time than the CCIS packet. On the otherhand, there is a high likelihood that only one packet will be affected since packets are separated by a long time interval (343 msec) relative to the burst interval.

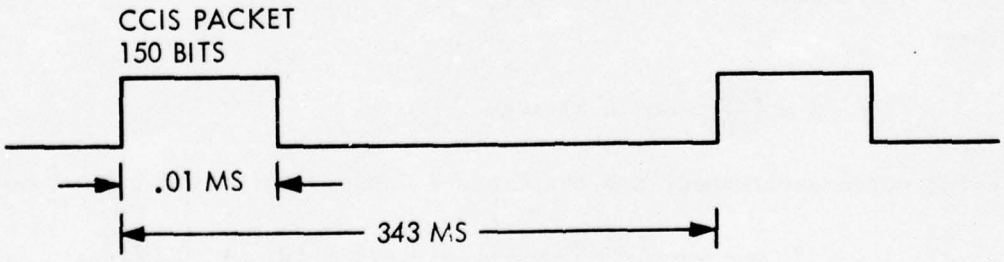


1 Hz RATE - BURSTS OCCUR ONCE EVERY SECOND



20 Hz RATE - BURSTS OCCUR ONCE EVERY 1/20 SECOND

(A) BURST ERROR RATES



(B) CCIS PACKET RATE

9379-75E

Figure 10-7. CCIS Message Error Environment

We now consider the effect of error environment on CCIS message transmission. In Table 10-11, equation 3, an expression for the optimum efficiency ratio Eff_R is given for Continuous ARQ error control with retransmission of the erroneous packet only. Since we are not sending CCIS messages at the optimum packet size, but at an average packet size of 150 bits/packet, the opt does not apply. The applicable equation is then

$$Eff_R = \frac{\frac{I}{P}\left(\frac{1-DC}{2-DC}\right)e^{-PE_B}}{1 - \left(\frac{1-DC}{2-DC}\right)e^{-PE_B}}$$

where DC = Duty cycle of error burst = .05

P = I + C = packet length

I = Information bits = 150

C = Control bits = 0

E_B = Bit error rate = .001 (worst case).

In our calculation, all the bits of the CCIS message are assumed to be overhead bits. Therefore,

$$Eff_R = \frac{\left(\frac{.95}{1.95}\right)e^{-.15}}{1 - \left(\frac{.95}{1.95}\right)e^{-.15}} = .7217$$

and $$\frac{1}{Eff_R} = 1.3856$$

Because of ARQ retransmission due to noise the equivalent CCIS Class I transmission rate becomes

$$436.73 \times \frac{208}{150} = 605.6 \text{ bits/sec}$$

in the worst noise environment and the Class I inefficiency (i.e., $1 - E_{ff}$) becomes

$$1 - Eff_I = \frac{605.5}{605.6 + 2.45 \times 10^6} = 2.471 \times 10^{-4} = .0002471$$

or approximately .025 percent. Therefore, under worst case conditions, (i.e., worst case error environment (10^{-3} BER)) and radial link traffic load, the CCIS traffic adds a negligible amount of transmission overhead.

10.2.3.2.2.2.2 Class II CCIS Messages (Packets)

The transmission overhead calculations for Class II CCIS Messages are similar to that performed for Class I messages. Table 10-8 shows the sequence of CCIS messages required to complete a Class II call and these are illustrated in Figure 10-8. The average CCIS message will be taken to be 150 bits. The single spoked wheel configuration is used as the network structure and the radial link is taken as a worst case traffic load. The portion of the link used for Class II traffic is

$$(10.1 \times 10^9)\ (.126) = 1.2726 \times 10^9 \text{ bits/BH}$$

From Table A-2, DAX Traffic Statistics, the average message length calculates to be

$$\frac{44.4046 \times 10^9 \text{ bits/BH}}{1{,}624{,}828 \text{ msg's/BH}} = 27{,}330 \text{ bits/msg}$$

and the number of Class II call originations per busy hour is

$$\frac{\text{call-orig}}{\text{BH}} = \frac{1{,}2726 \times 10^9 \text{ bits/BH}}{27{,}330 \text{ bits/msg}} = 46{,}464. \ *$$

Call originations per busy second are

$$\frac{\text{call-orig}}{\text{sec}} = \frac{46{,}564}{3600} = 12.934.$$

The number of CCIS messages per call is 6; therefore

$$\text{Messages/sec} = (12.934)\ (6) = 77.604$$

and

$$\text{Bits/sec} = (77.604)\ (150) = 11640.6.$$

---

* It is probable that several messages will be transmitted during a call initiation instead of one message, as is assumed here. Hence, the number of call originations would be considerably less than 46,564. We have chosen the above approach as a worst case in order to demonstrate that the transmission overload generated is not significant.

TABLE 10-8

CCIS MESSAGE SEQUENCE FOR COMPLETING CLASS II CALLS

TANDEM SWITCH

MESSAGE #	Message	# of bits/packet
1	Call Initiate Packet to Next Switch	232
2	Acknowledge to Preceeding Switch	104
3	Answer to Preceeding Switch	168
4	Acknowledge to Next Switch	104
5 *	Call Release to Next Switch	168
6	Acknowledge to Preceeding Switch	104
		880 bits
		Avg = 146.7

ORIGINAL SWITCH

1	Call Initiate Packet to Next Switch	232
4	Acknowledge to Next Switch	104
5 *	Call Release to Next Switch	168
		504
		Avg = 168

TERMINATING SWITCH

2	Acknowledge to Preceeding Switch	104
3	Answer to Preceeding Switch	168
5 *	Acknowledge to Preceeding Switch	104
		Avg = 125.3

* Assume that the called subscriber goes on-hook first.

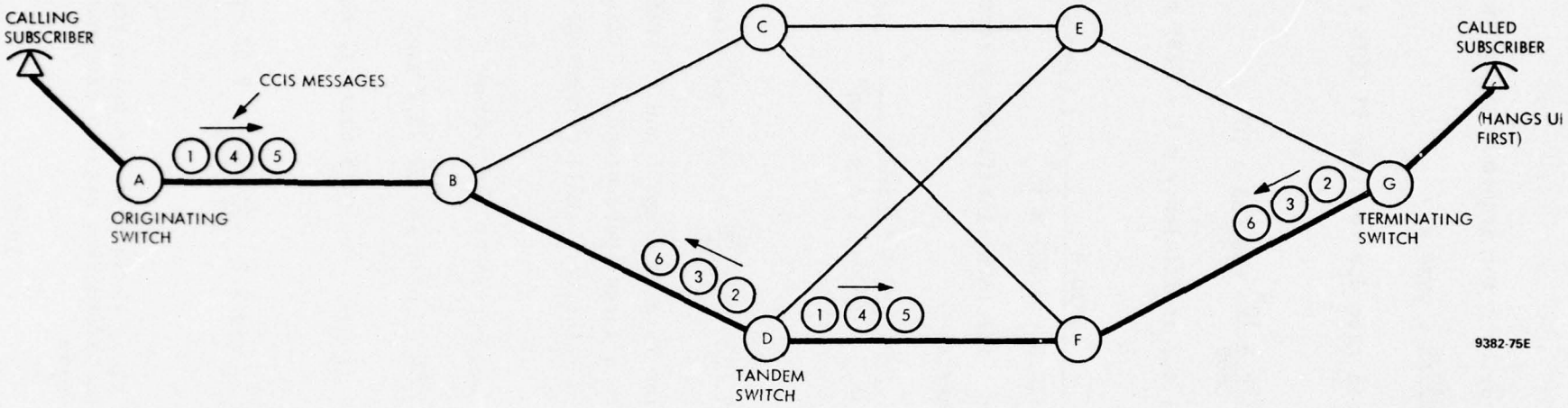


Figure 10-8. CCIS Messages Transmitted During Typical Class II Call Sequence (See Table 10-8 for Definition of Messages)

Taking into account the factor of 2 for transmitting on two links,

$$\text{Bits/sec} = \frac{11640.6}{2} = 5820.3$$

is the equivalent transmission rate due to Class II CCIS traffic. The equivalent rate of Class # traffic is

$$\text{Bits/sec} = \frac{1.276 \times 10^9}{3600} = 3.535 \times 10^5 \ .$$

The transmission overhead (i.e., inefficiency = 1 - Eff) relating to Class II CCIS messages is

$$1 - \text{Eff}_{II} = \frac{5820.3}{5820.3 + 3.535 \times 10^5} = 1.6198 \times 10^{-2} = .016$$

or approximately 1.6 percent. The total inefficiency represented by CCIS Class I *and II messages taken together* is

$$1 - \text{Eff}_T = \frac{5280.3 + 436.73}{5280.3 + 436.73 + 2.8 \times 10^6} = .00204$$

or approximately .2%, where

$$\frac{10.1 \times 10^9 \text{ bits/BH}}{600} \ \frac{\text{bits/BH}}{\text{sec/BH}} = 2.8 \times 10^6 \text{ bits/sec}$$

is the equivalent transmission rate of Class I and II traffic on the radial link. Thus CCIS messages constitute a larger percentage of Class II overhead than Class I overhead, however, they are a relatively small percentage of the total Class II overhead * .

A Class II CCIS Message occurs on the average

$$\frac{150 \text{ bits/CCIS msg}}{5820.3 \text{ bits/sec}} = .0284 \text{ sec or } 28.4 \text{ msec}$$

or 1 CCIS message every 2.84 frames. One CCIS Class II call origination and termination occurs every

$$2.84 \text{ frames/CCIS msg} \times 3 \ \frac{\text{msg's}}{\text{call orig}} = 8.52 \text{ frames.}$$

---

* In subsequent sections it is shown that the major source of Class II transmission overhead is the non-information bits in each packet and the need for retransmission due to noise.

For a 10^{-3} BER and Continuous ARQ error control as with Class I CCIS messages, the equivalent CCIS required data rate becomes

$$5280.3 \times \frac{208}{150} = 7322.0 \text{ bits/sec.}$$

Thus the overhead becomes

$$1 - \text{Eff}_{II} = \frac{7322.0}{7322.0 + 3.535 \times 10^5} = .02029$$

or approximately 2.03%. The total efficiency of Class I & II CCIS messages under worst case error and traffic conditions is

$$1 - \text{Eff}_T = \frac{605.6 + 7322.0}{605.5 + 7322.0 + 2.8 \times 10^6} = .00272$$

or approximately .27 percent. Thus, CCIS traffic is a relatively small portion of the traffic, even under worst case error and traffic conditions.

10.2.3.2.2.2.3 Automatic Message Accounting (AMA)

AMA messages will be generated everytime a call connection is made. They provide journal entries on call duration, calling and called party, identification precedence, and any other data deemed desirable to record for future access. We will assume that AMA will happen every time a call is originated and will consist of three messages, each 150 bits long, for a total of 450 bits. Under these conditions 14 messages are generated for every Class I call instead of 11 and 9 messages are generated for every Class II call instead of 6. The new efficiency becomes

$$\text{Eff}_T = \frac{5280.3\ (9/6) + 436.73\ (14/11)}{7920.5 + 555.08 + 2.8 \times 10^6} = .0030 \text{ or } .3 \text{ percent}$$

With 10^{-3} BER, $E_{ffT} = .0042$ or .42%, and one CCIS message (Class I or II) occurs every 1.28 frames. Therefore taking everything into account, it seems safe to say that the CCIS messages will not be a significant source of transmission overload in the DAX network.

10.2.3.2.3 Transmission Efficiency for Class II Traffic

In the previous section it was shown that the overhead due to CCIS traffic represents a very small percentage of the transmission capacity of the SENET-DAX trunk.

In contrast the major loss of Class II transmission capability is due to the provisions required for error control of the data. The other source of Class II transmission overhead is the non-information bits which have to be transmitted in each packet for message and destination identification. The magnitude of Class II transmission efficiency is a complex function of error environment, packet size and structure and the type of ARQ used. In the next section transmission efficiency equations are derived for each type of ARQ and maximized for packet size.

10.2.3.2.4 Optimization of Transmission Efficiency for Class II Traffic

10.2.3.2.4.1 Designation of Parameters

The following parameters which are used in the equations of transmission efficiency that are derived and optimized in this section are to be interpreted as defined below.

10.2.3.2.4.1.1 Channel or Trunk Parameters

R = The number of bits transmitted over the channel or trunk per unit time.

E_b = probability of a bit being incorrect when transferred across the channel if the channel is usable.

The channel will have noise bursts during which its error rate is so high that it is considered non-usable.

DC = duty cycle of noise burst (i.e., percentage of time during which burst condition exists).

1-DC = probability of channel being usable.

10.2.3.2.4.1.2 Packet Switching System Parameters

Assume a packet switching system having the following characteristics:

1) Information is transferred across the channel in packets containing P bits. Where

$$P = I + C \qquad (1)$$

2) The P bits are made up of I information bits and C control bits.

3) An ARQ system is used for error control.

The receiver terminal monitors the incoming signal for transmission errors and notifies the transmitter by means of a control message sent over a return link as to whether the packet need be repeated.

The transmitter may operate in a block by block mode or in one of two continuous modes. The two continuous modes differ in that either all data occurring after a NAK is retransmitted or only the errored packet.

10.2.3.2.4.1.3 Figure of Merit - Transmission Efficiency (E_{ff})

The figure of merit of the switching system is the transmission efficiency of the channel (E_{ff})

$$E_{ff} = ITR/R \tag{2}$$

ITR = rate of transfer of correct information

R = channel transmission rate

10.2.3.2.4.2 Optimum Packet Size in Block by Block Mode of ARQ

In a block by block mode the transmitter remains idle from the time it sends the last bit of a packet until it receives and interprets the return ACK/NAK control message. Let W equal this waiting time per packet.

$$E_{ff} = \frac{I(1-E_p)}{P} \cdot \frac{T \cdot (1-DC)}{T+W} \tag{3}$$

$$T = \frac{P}{R(1-DC)} \tag{4}$$

for a random error pattern.

$$E_p = 1 - (1-E_B)^P \tag{5}$$

for small values of E_p

$$1 - E_p = (1 - E_B)^P \cong e^{-PE_B} \tag{6}$$

therefore

$$E_{ff} = \frac{I \cdot e^{-PE_B}}{P} \quad \frac{\frac{P}{R}}{\frac{P}{R(1-DC)} + W} \quad \frac{I \cdot e^{-PE_B}(1-DC)}{I + C + RW(1-DC)} = \frac{I \cdot e^{-PE_B}}{I + K} \tag{7}$$

where $K = WR(1-DC) + C$ (8)

$$\frac{d(E_{ff})}{dP} = E_{ff}\left[\frac{1}{I} - \frac{1}{I + WR(1-DC) + C} - E_B\right] = 0 \tag{9}$$

Then the optimum value of I is given by the solution of

$$I^2 + KI - \frac{K}{E_B} = 0 \tag{10}$$

$$I_{opt} = \frac{K}{2}\left(\sqrt{(1 + \frac{4}{KE_B})} - 1\right) \tag{11}$$

from equation (1) we get

$$P_{opt} = I_{opt} + C \tag{12}$$

10.2.3.2.4.2.1 Waiting Time (W)

From the formula for E_{ff} given in equation (7) it is seen that E_{ff} is optimized by minimizing K. This is done by minimizing W and C which are parameters of the switching system.

The waiting period (W) is the sum of the following elements:

W_{pf} = forward propagation time across the channel - a function of the channel

W_{PROC-R} = delay for receiver processor to determine accuracy of packet data and format return control packet having P_c bits

T_{tr} - time to transmit control packet

W_{pr} = reverse channel propagation time

W_{proc-T} = time for transmitting computer to decode the control packet and schedule the next packet for transmission.

The two processing delays, (W_{PROC-R} and W_{PROC-T}) and the length of the return control packet (P_c) are the parameters which depend upon the elements of the packet switch design relating to error control, message and network control, and synchronization.

10.2.3.2.4.2.2 Control Packet Transmission Time (W_{T_r})

The value of W_{tr} is dependent upon P_c and is given by the following formula:

$$W_{tr} \tilde{=} \frac{P_c}{R(1-DC)} e^{P_c E_B} + P_c E_B T_r \qquad (13)$$

The first term on the right covers the normal transmission time plus any infrequent repeats. The value of P_c is small, so it is assumed that $P_c E_B << 1$. There is no acknowledgement of the ACK/NAK message. Instead the control block is retransmitted after a time, T_r, if there is no intervening signals from the transmitting terminal. The second term on the right is the expected value of additional waiting time due to time-outs of the ARQ control block at the transmitting terminal.

10.2.3.2.4.2.3 Receiver Processing Time (W_{PROC-R})

The receiver processing time is dependent upon the computer hardware and software and the error detection procedure. From examination of equation (7) it is seen that increases in W result effectively in a decrease in the period during which the channel can transmit required information bits in much the same manner) as an increase in control bits (c). Normally algebraic and/or convolutional coding procedures are considered more powerful than simple horizontal and vertical parity checks due to the fact that the same error detection capability can be attained with less overhead bits. From the viewpoint of maximizing transmission efficiency, however, the optimum error decoding procedure is the one which minimizes $W_{PROC-R} + C$. This may be attained by the procedure with the most rapid decision procedure for a given detection capability rather than the minimum C.

10.2.3.2.4.2.4 Non Information (overhead) bits (C)

The non-information bits (c) per packet consist of:

(1) Frame and classmark sequences
(2) Flag sequence
(3) Address fields
(4) Control fields
(5) Frame check sequence and/or parity bits
(6) Stuffing bits for filling out computer words or transmission blocks
(7) Packet length and message service or identification features.

10.2.3.2.4.3 Optimum Packet Size for Continuous Modes of ARQ

There are two types of continuous ARQ to be considered. Type I is where all data after the receipt of a NAK is retransmitted. Type II, on the contrary, is where only the single errored block is retransmitted. Continuous ARQ is only possible on a full duplex link such as will be available in the proposed DAX network. Either continuous ARQ mode produces far better transmission efficiency than block by block but both require greater data storage capacity. Repeat of only the NAK packet produces the greatest improvement in efficiency in the noisy environments but requires both greater storage capacity and a more complex packet control program.

10.2.3.2.4.3.1 Optimum Packet Size for Continuous ARQ Type I

Assume a protocol which calls for retransmission of all data since the packet that reported a NAK. In this case E_{ff} is given by

$$E_{ff} = \frac{I \cdot e^{-PE_B}(1-DC)}{I + C + WR\,(1-DC)(1-e^{-PE_B})} = \frac{I \cdot e^{-PE_B}(1-DC)}{I + K - WR\,(1-DC)\,e^{-PE_B}} \qquad (14)$$

eliminating P and I variation from the numerator we get

$$E_{ff} = \frac{(1-DC)}{\left(1 + \frac{K}{I}\right)e^{PE_B} - \frac{WR\ (1-DC)}{I}} = \frac{(1-DC)}{D} \tag{15}$$

E_{ff} is maximized by minimizing the denominator (D). Remembering that $P = I + C$ we differentiate D with respect to I and equate to zero to obtain I opt. I opt is the solution of:

$$\frac{-K}{I^2} e^{PE_B} + \frac{I + K}{I} E_B e^{PE_B} + \frac{WR\ (1-DC)}{I^2} = 0 \tag{16}$$

simplifying we get

$$E_B I_{opt}^2 + KE_B I_{opt} - K + (K-C)\ e^{-P_{opt}E_B} = 0 \tag{17}$$

assume: $K = 8860.8$; $C=60$; $E_B = .001$; $DC = .05$ for our example

$$.001\ I^2 + 8.8606\ I - 8860.8 + 8800.8\ e^{-.001P} = 0 \tag{18}$$

Solving by successive approximation, using the Newton-Raphson algorithm, we get

I opt = 292 bits

P_{opt} = 352 bits

$E_{ff\ opt}$ = .06678

10.2.3.2.4.3.2 Optimum Packet Size for Continuous ARQ Type II

Assume protocol calls for retransmission of the error packet alone then E_{ff} is given by:

$$E_{ff} = \frac{I \cdot e^{-PE_B}\ (1-DC)}{I + C + P(1-DC)(1-e^{-PE_B})} \tag{19}$$

$$E_{ff} = \frac{(1-DC)}{e^{PE_B} + \frac{C}{I} e^{PE_B} + \frac{P(1-DC)}{I} e^{PE_B} - \frac{P(1-DC)}{I}} \tag{20}$$

differentiating the denominator and equating to zero we get

$$e^{PE_B} E_B - \frac{C}{I^2} + \frac{CE_B}{I} + \frac{(1-DC)}{I} - \frac{P(1-DC)}{I^2} + \frac{P(1-DC)E_B}{I} + \frac{P(1-DC)}{I^2} - \frac{(1-DC)}{I} = 0 \quad (21)$$

Simplifying:

$$E_B I^2 + I\,(CE_B + (1-DC) + P\,(1-DC)\,E_B) - C - P\,(1-DC) + e^{-PE_B}(1-DC)(C) = 0 \quad (22)$$

setting $P = I + C$ we get

$$I^2(E_B)(2-DC) + I(CE_B)(2-DC) - C(2-DC) + e^{-PE_B}(1-DC)C = 0$$

Solve for I_{opt} using the Newton Raphson algorithm with

$E_B = .001$; $DC = .05$; $C = 60$

$$.00195\ I^2 + .117I - 117 + 57\ e^{-PE_B} = 0$$

we get:

$I_{opt} = 164$ bits

$P_{opt} = 224$

$E_{ff\ opt} = .467$

Using typical baseline DAX parameter values, transmission efficiency and other performance parameters were computed for varying values of channel bit error rate (E_B) for Block By Block continuous with total retransmission and continuous with single packet retransmission. The results are tabulated in Tables 10-9, 10-10 and 10-11.

Figures 10-9, 10-10 and 10-11 are graphs of transmission efficiency versus error rates varying from 10^{-3} to 10^{-7} for various packet sizes and the three forms of ARQ.

Tables 10-12, 10-13 and 10-14 show the variation of transmission efficiency versus packet size for various bit error rates for the three types of ARQ. They show the sensitivity of transmission efficiency to packet size variation. Figures 10-12, 10-13 and 10-14 graphically display the same information.

TABLE 10-9. OPTIMUM TRANSMISSION EFFICIENCY vs E_B FOR BLOCK BY BLOCK ARQ

E_B	I_{opt}	P_{opt}	$E_{ff,opt}$	$E_{Noise} = EII/E_{ff,opt}$	$E_p = 1-\ell^{-P_{opt}E_B}$	$E_{II\ MAX}$ BUSY HOUR	$E_{II\ MAX}$ E_I = 20 Erlangs	$\rho II = \frac{E_{Noise}}{\text{BUSY HR } 20}$	$\rho_{II} = \frac{E_{Noise}}{E_I = 20\ 80}$
10^{-3}	907	967	.0335	119.26	.620	.6044	.652	Overload	Overload
10^{-4}	5973	6033	.2092	19.12	.453	3.766	5.420	.956	.664
10^{-5}	25,665	25,725	.5460	7.326	.227	9.828	22.28	.366	.162
10^{-6}	89,806	89,865	.7904	5.061	.086	14.23	44.894	.253	.080
10^{-7}	293,274	293,333	.8955	4.467	.029	16.119	59.29	.2233	.061

E_B = Bit Error Rate

I_{opt} = Optimum number of information bits in a packet

P_{opt} = Optimum number of bits in a packet

$E_{ff,opt}$ = Optimum transmission efficiency

E_{NOISE} = Peak data load in noise environment

E_p = Packet error rate

R_{II} = Class II data carrying capacity of trunk

(1) $$I_{opt} = \frac{K}{2}\sqrt{\left(1 + \frac{4}{KE_B}\right)} - 1$$

(2) $$K = C + WR_{II}\ (1\text{-}DC)$$

(3) $$E_{ff,opt} = \frac{I_{opt} \cdot \ell^{-P_{opt}E_B}\ (1\text{-}DC)}{I_{opt} + C + WR_{II}\ (1\text{-}DC)}$$

(4) $$R_{II} = R\left(\frac{CH - E_I}{CH}\right)$$

$E_{II\ MAX\ (busy\ hour)}$ = Peak data load in busy hour for 90% data utilization factor

$E_{II\ MAX\ (F_I = 20)}$ = Peak data load when E_{II} = 20 Erlangs for 90% data utilization

ρ_{II} = Data utilization factor

C = Non information bits = 60 bits

W = Acknowledgment delay = .03 seconds

R = Transmission rate of trunk = 1.544 mb/s

E_I = 80 Erlangs

CH = 100 channels at 15,400 bits/sec

K = 4 Erlangs

DC = Burst duty cycle = .05

E_{II} = 4 Erlangs

TABLE 10-10. OPTIMIZATION OF E_{ff} FOR CONTINUOUS ARQ REPEATING ALL DATA FROM NAK

E_B	I_{opt}	P_{opt}	$E_{ff,opt}$ OPTIMUM TRANSMISSION EFFICIENCY	$E_{Noise} = E_{II}/E_{ff}$	$E_\rho = 1-e^{-PE_B}$ Packet error Rate	$E_{II\ MAX}$ Busy Hour E_I = 80 Erlangs	$E_{II\ MAX}$ Slow Traffic E_I = 20 Erlangs	$\rho_{II} = E\ \frac{Noise}{20}$ Busy Hour Channel utilization factor	$\rho_{II} = E\ \frac{Noise}{80}$ Slow Hour
10^{-3}	292	352	.0658	60.76	.297	1.185	1.301	Overload	Overload
10^{-4}	850	910	.4400	9.09	.087	7.92	13.36	.4545	.2694
10^{-5}	2470	2530	.8320	4.808	.025	14.976	48.315	.2404	.0745
10^{-6}	7732	7792	.9272	4.314	.0078	16.690	65.065	.2157	.0553
10^{-7}	24,450	24,510	.9445	4.235	.0024	17.001	67.827	.2117	.0531

E_B = Bit error Rate

I_{opt} = Optimum Information bits/packet

P_{opt} = Optimum Total bits/packet

$E_{ff,opt}$ = Optimum transmission efficiency

E_{Noise} = Peak data load in noise environment

E_p = Packet error rate

R_{II} = Class II data capacity of trunk

ρ = Utilization factor

$E_{II,MAX}$ = Peak data load in Erlangs for 90% utilization factor

E_I = Peak voice load in Erlangs = 80 Erlangs

E_{II} = Peak hour equivalent data load = 4 Erlangs

R = Trunk modem rate = 1.544 MBS for T_1

W = Acknowledgement delay = .03 seconds for 1500 mile trunk

CH = Channel capacity = 100

K = 8860.8 bits during peak voice traffic

DC = Duty cycle of error bursts = .05

C = Non information bits/packet = 60 bits

(1) I_{opt} is solution of

$$E_B I_{opt}^2 + KE_B I_{opt} - K + (K-C)\,\ell^{-P_{opt}E_B} = 0$$

(2) $$K = C + WR_{II}(1-DC)$$

(3) $$P_{opt} = I_{opt} + C$$

(4) $$E_{ff,opt} = \frac{I_{opt} \cdot \ell^{-P_{opt}E_B} \cdot (1-DC)}{I_{opt} + K - WR_{II}(1-DC)\ell^{-P_{opt}E_B}}$$

(5) $$R_{II} = R\left(\frac{CH-E_I}{CH}\right)$$

TABLE 10-11. OPTIMIZATION OF E_{ff} FOR CONTINUOUS ARQ REPEATING ONLY NAK PACKET

E_B	I_{opt}	$P_{opt} = I_{opt} + C$	$E_{ff'opt}$	$E_{Noise} = E_{II}/E_{ff}$	$E_\rho = 1 - \ell^{-P_{opt}E_B}$ Packet error rate	E_{II}, MAX Busy hour E_I = 80 Erl	E_{II}, MAX Slow hour E_I = 20 Erl	$\rho_{II,MAX}$ Busy hour	$\rho_{II,MAX}$ Slow hour
10^{-3}	164	224	.467	8.565	.2007	8.406 Erl	33.624	.4275	.1069
10^{-4}	540	600	.763	5.242	.0582	13.734	54.936	.2621	.0655
10^{-5}	1738	1798	.8869	4.51	.0178	15.96	63.84	.2255	.0564
10^{-6}	5526	5586	.9296	4.303	.0056	16.73	66.92	.2152	.0538
10^{-7}	17,506	17,566	.9435	4.240	.0018	16.98	67.92	.212	.053

Same definitions and parameter values as Table 8

(1) I_{opt} is a solution of $I_{opt}^2 E_B + I \cdot C \cdot E_B - C + \frac{1-DC}{2-DC} C\ell^{-\rho E_B} = 0$

(2) $P_{opt} = I_{opt} + C$

(3) $E_{ff,opt} = \frac{I_{opt}}{P_{opt}} \cdot \left(\frac{1-DC}{2-DC}\right) \cdot \ell^{-P_{opt}E_B} \Big/ \left(1 - \left(\frac{1-DC}{2-DC}\right)\ell^{-P_{opt}E_B}\right)$

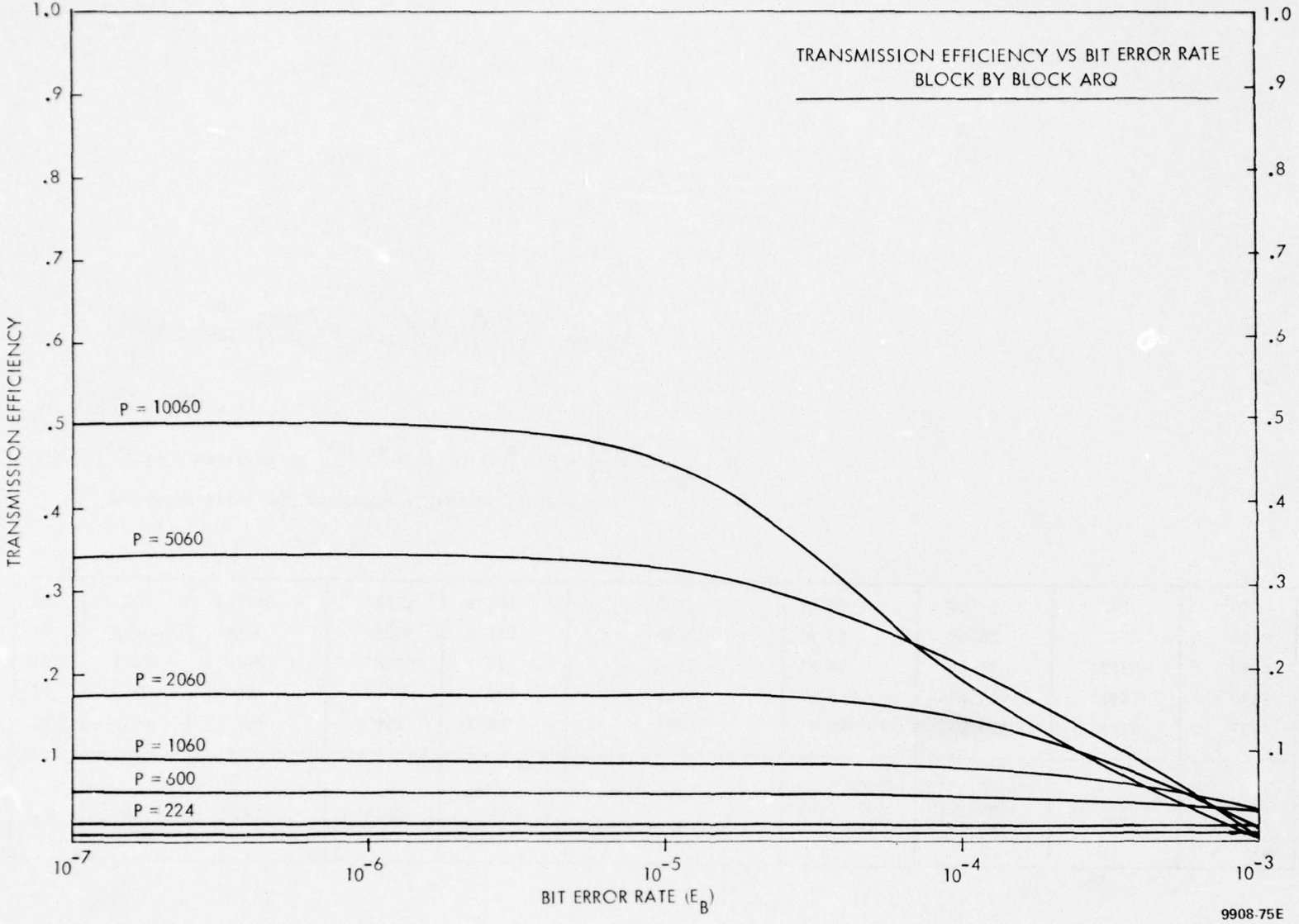


Figure 10-9. Transmission Efficiency vs Bit Error Rate Block by Block ARQ

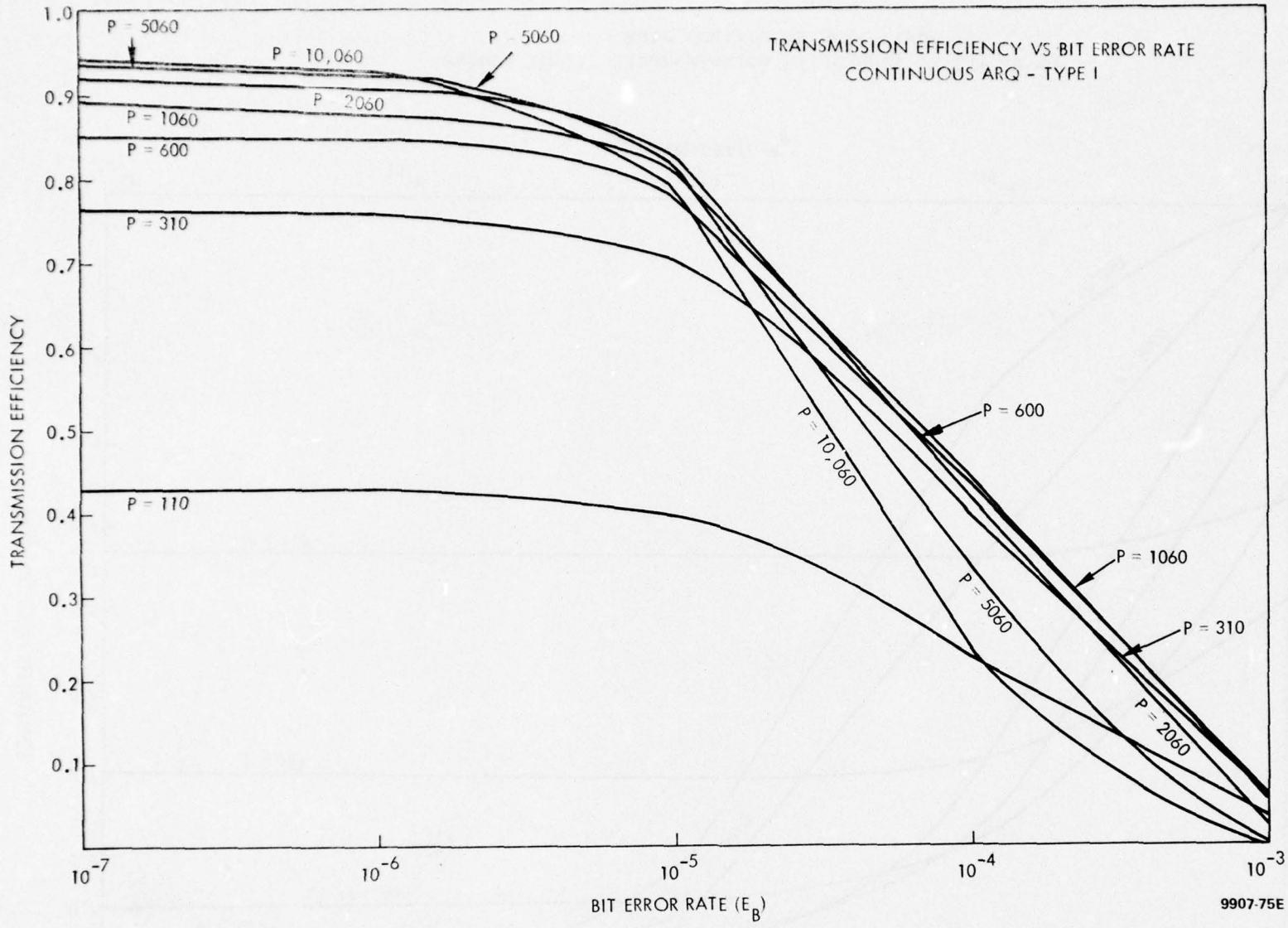


Figure 10-10. Transmission Efficiency vs Bit Error Rate Continuous ARQ - Type 1

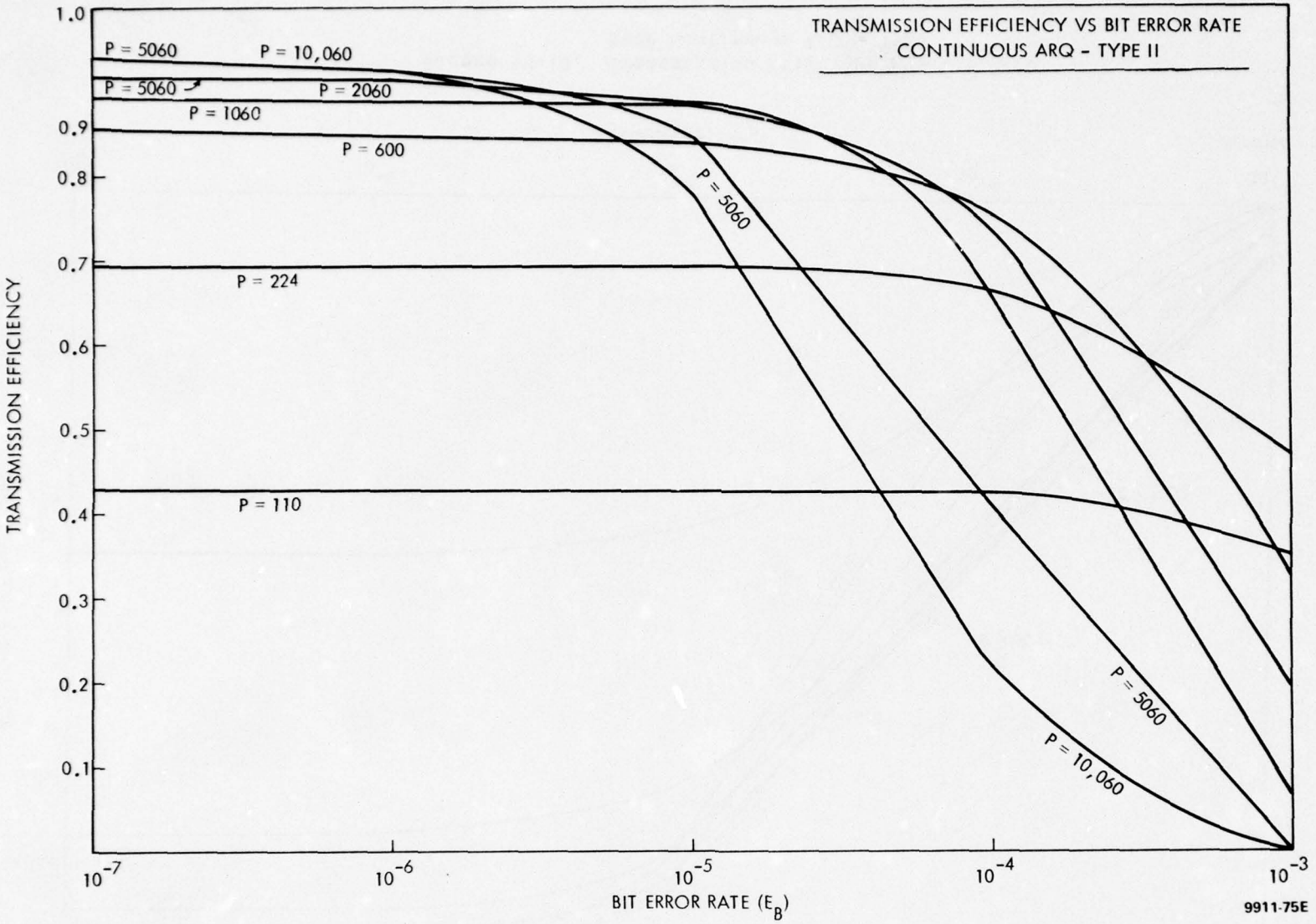


Figure 10-11. Transmission Efficiency vs Bit Error Rate Continuous ARQ - Type II

TABLE 10-12. TRANSMISSION EFFICIENCY (E_{ff}) VERSUS PACKET SIZE (P) AND BIT ERROR RATE (E_B) FOR BLOCK BY BLOCK ARQ

INFORMATION BITS PER PACKET I	TOTAL BITS PER PACKET P	TRANSMISSION EFFICIENCY (E_{ff})* $E_B=10^{-3}$	$E_B=10^{-4}$	$E_B=10^{-5}$	$E_B=10^{-6}$	$E_B=10^{-7}$
50	110	.0048	.0053	.0053	.0053	.0053
100	160	.0090	.0109	.0106	.0106	.0106
150	210	.0128	.0155	.0158	.0158	.0158
164	224	.0138	.0169	.0172	.0173	.0173
200	260	.0162	.0204	.0209	.0210	.0210
250	310	.0191	.0253	.0260	.0261	.0261
300	360	.0217	.0300	.0310	.0311	.0311
350	410	.0240	.0346	.0360	.0361	.0361
400	460	.0259	.0392	.0408	.0410	.0410
450	510	.0276	.0436	.0457	.0459	.0459
500	560	.0290	.0480	.0505	.0507	.0507
540	600	.0299	.0514	.0542	.0545	.0546
600	660	.0311	.0567	.0594	.0602	.0602
700	760	.0325	.0645	.0690	.0695	.0695
800	860	.0333	.0722	.0780	.0786	.0787
900	960	.0335	.0796	.0868	.0875	.0876
1,000	1,060	.0334	.0867	.0953	.0962	.0963
1,500	1,560	.0289	.1177	.1354	.1373	.1375
2,000	2,060	.0223	.1424	.1714	.1746	.1749
2,500	2,560	.0162	.1618	.2038	.2085	.2090
3,000	3,060	.0113	.1769	.2330	.2396	.2402
3,500	3,560	.0076	.1884	.2596	.2680	.2689
4,000	4,060	.0051	.1969	.2837	.2943	.2954
4,500	4,560	.0033	.2028	.3057	.3185	.3198
5,000	5,060	.0022	.2066	.3258	.3410	.3425
5,500	5,560	.0019	.2087	.3442	.3618	.3636
6,000	6,060	.0009	.2092	.3610	.3812	.3833
6,500	6,560	.0006	.2086	.3765	.3994	.4017
7,000	7,060	.0004	.2070	.3907	.4163	.4190
7,500	7,560	.0002	.2045	.4038	.4322	.4352
8,000	8,060	.0001	.2013	.4158	.4471	.4504
8,500	8,560	.0001	.1976	.4270	.4612	.4647
9,000	9,060	.0001	.1935	.4372	.4744	.4783
9,500	9,560	.0000	.1890	.4467	.4869	.4911
10,000	10,060	.0000	.1842	.4555	.4986	.5032

* $E_{ff} = \frac{I(1-DC)(e^{-PE_B})}{P + WR(1-DC)}$; W is delay in receipt of ACK/NAK; R is effective data rate of trunk used for Class II traffic; DC is burst duty cycle. In the above table

W = .03 seconds R = 308.800 b/s and DC = .05

TABLE 10-13. TRANSMISSION EFFICIENCY (E_{ff}) VS PACKET SIZE (P) AND BIT ERROR RATE (E_B) FOR CONTINUOUS ARQ TYPE 1 *

Information Bits per Packet I	Total BITS per Packet P	Transmission Efficiency (E_{ff})** $E_B = 10^{-3}$	$E_B = 10^{-4}$	$E_B = 10^{-5}$	$E_B = 10^{-6}$	$E_B = 10^{-7}$
50	110	.0414	.2278	.3965	.4280	.4314
100	160	.0554	.3120	.5449	.5901	.5934
150	210	.0615	.3552	.6224	.6725	.6780
164	224	.0626	.3636	.6379	.6893	.6949
200	260	.0644	.3810	.6700	.7242	.7301
250	310	.0656	.3979	.7021	.7592	.7654
300	360	.0658	.4096	.7251	.7855	.7910
350	410	.0654	.4180	.7424	.8036	.8102
400	460	.0647	.4241	.7559	.8185	.8253
450	510	.0638	.4287	.7667	.8305	.8375
500	560	.0626	.4321	.7754	.8403	.8467
540	600	.0616	.4343	.7816	.8470	.8542
600	660	.0600	.4366	.7888	.8555	.8628
700	760	.0571	.4390	.7984	.8667	.8742
800	860	.0542	.4399	.8056	.8753	.8829
900	960	.0512	.4399	.8111	.8820	.8898
1000	1060	.0483	.4392	.8154	.8875	.8953
1500	1560	.0352	.4306	.8301	.9041	.9125
2000	2060	.0249	.4181	.8311	.9124	.9213
2500	2560	.0172	.4043	.8320	.9173	.9267
3000	3060	.0117	.3901	.8313	.9204	.9303
3500	3560	.0070	.3759	.8296	.9226	.9328
4000	4060	.0052	.3619	.8274	.9241	.9348
4500	4560	.0034	.3481	.8248	.9251	.9362
5000	5060	.0022	.3348	.8219	.9259	.9374
5500	5560	.0014	.3217	.8188	.9264	.9384
6000	6060	.0009	.3091	.8156	.9268	.9392
6510	6560	.0006	.2969	.8123	.9270	.7399
7000	7070	.0004	.2850	.8090	.9272	.9404
7500	7560	.0002	.2736	.8056	.9272	.9409
8000	8060	.0001	.2625	.8021	.9272	.9413
8500	8560	.0001	.2518	.7986	.9272	.9417
9000	9060	.0001	.2416	.7951	.9271	.9420
9500	9560	.0000	.2316	.7915	.9269	.9423
10000	10060	.0000	.2221	.7880	.9268	.9426

* Continuous ARQ type I is defined as continuous transmission with retransmission of all data after the errored packet.

** $E_{ff} = \dfrac{I\,\ell^{-PE_B}(1-DC)}{P + (1-DC)\,W \cdot R_{II}\,\ell^{-PE}}$

W = waiting time for Ack

R_{II} = effective data capacity of trunk

TABLE 10-14. TRANSMISSION EFFICIENCY (E_{ff}) VS PACKET SIZE (P) AND BIT ERROR RATE (E_B) FOR CONTINUOUS ARQ TYPE II *

Information Bits per Packet I	Total BITS per Packet P	Transmission Efficiency (E_{ff}) ** $E_B = 10^{-3}$	$E_B = 10^{-4}$	$E_B = 10^{-5}$	$E_B = 10^{-6}$	$E_B = 10^{-7}$
50	110	.352	.4226	.4309	.4317	.4318
100	160	.4436	.5756	.5919	.5936	.5937
150	210	.4662	.6516	.6758	.6783	.6785
164	224	.4669	.6661	.6925	.6952	.6955
200	260	.4628	.6951	.7271	.7304	.7307
250	310	.4484	.7218	.7617	.7657	.7661
300	360	.4291	.7389	.7861	.7911	.7916
350	410	.4079	.7498	.8045	.8103	.8109
400	460	.3862	.7566	.8187	.8253	.8260
450	510	.3649	.7606	.8300	.8374	.8382
500	560	.3443	.7626	.8390	.8473	.8481
540	600	.3285	.7630	.8451	.8540	.8549
600	660	.3059	.7622	.8526	.8625	.8635
700	760	.2718	.7582	.8622	.8737	.8749
800	860	.2416	.7520	.8691	.8822	.8836
900	960	.2150	.7444	.8742	.8890	.8905
1000	1060	.1916	.7358	.8780	.8944	.8960
1500	1560	.1097	.6872	.8863	.9107	.9132
2000	2060	.0643	.6378	.8864	.9187	.9220
2500	2560	.0382	.5913	.8831	.9231	.9273
3000	3060	.0229	.5485	.8782	.9258	.9308
3500	3560	.0138	.5093	.8723	.9275	.9333
4000	4060	.0083	.4735	.8660	.9286	.9352
4500	4560	.0051	.4408	.8593	.9292	.9367
5000	5060	.0031	.4109	.8525	.9295	.9378
5500	5560	.0019	.3835	.8485	.9296	.9387
6000	6060	.0011	.3584	.8385	.9296	.9395
6500	6560	.0007	.3353	.8314	.9294	.9401
7000	7060	.0004	.3139	.8243	.9291	.9406
7500	7560	.0003	.2942	.8173	.9287	.9411
8000	8060	.0002	.2760	.8103	.9283	.9414
8500	8560	.0001	.2592	.8033	.9278	.9418
9000	9060	.0001	.2435	.7964	.9272	.9420
9500	9560	.00003	.2290	.7896	.9267	.9423
10000	10060	.00001	.2155	.7828	.9261	.9425

* Continuous ARQ type II is defined as continuous transmission with retransmission only of the errored packet on receipt of a NAK.

** $E_{ff} = \dfrac{I \cdot \frac{1-DC}{2-DC} \cdot e^{-PE_B}}{1 - \frac{1-DC}{2-DC} \cdot \ell^{-PE_B}}$

P = I + C
DC = Duty Cycle of Error Bursts
I = Information bits/packet

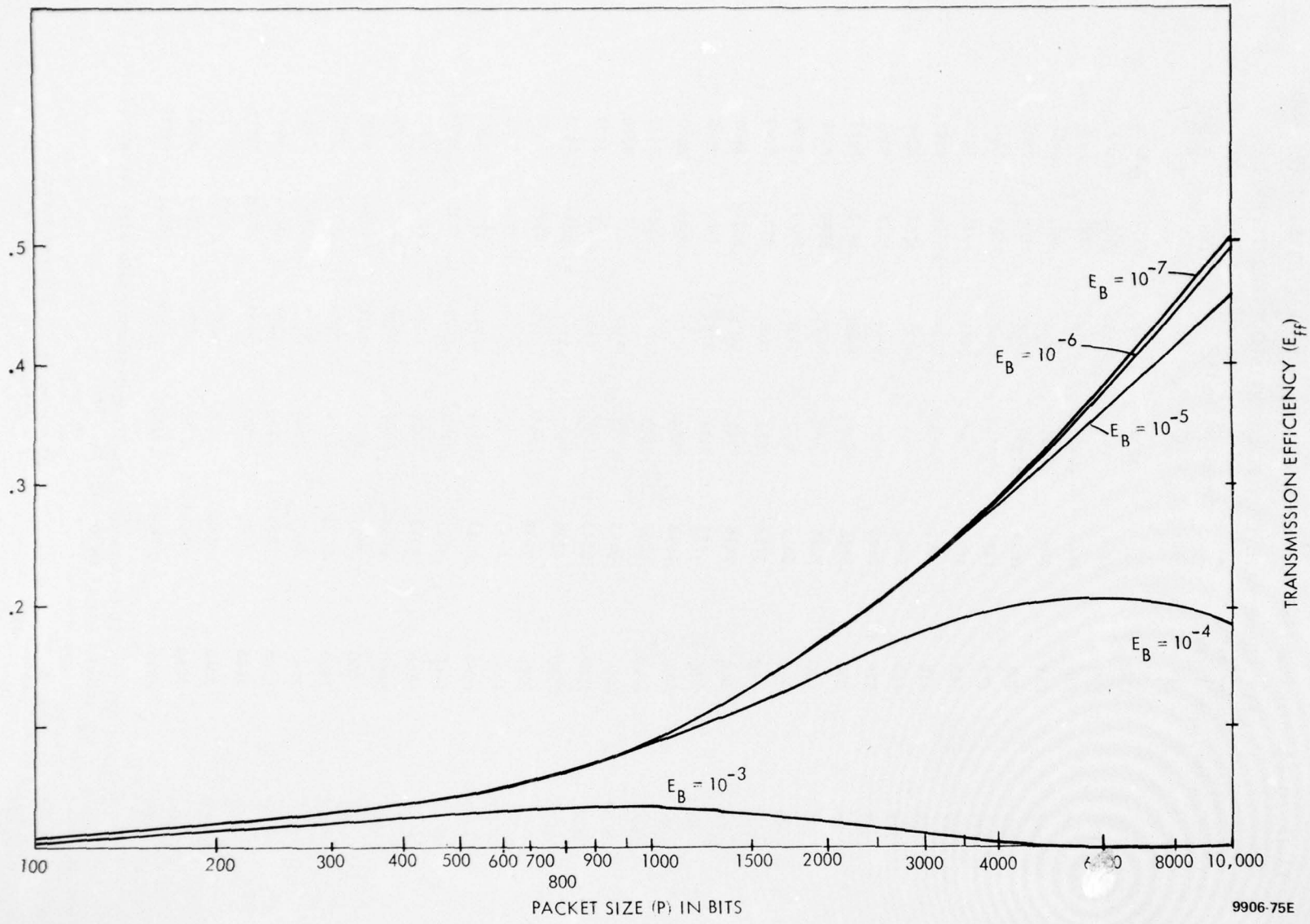


Figure 10-12. Transmission Efficiency vs Packet Size
Block by Block ARQ

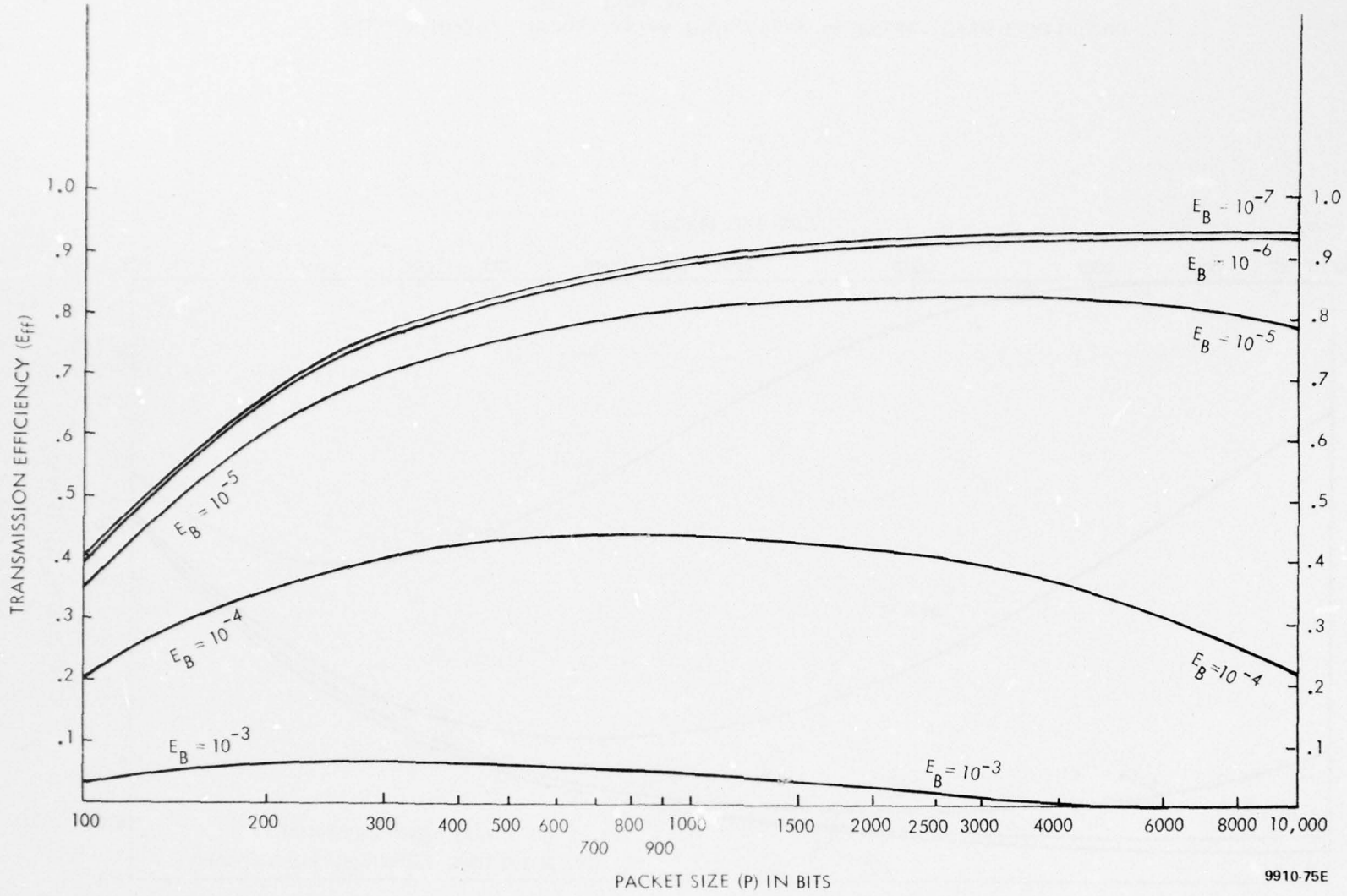


Figure 10-13. Transmission Efficiency vs Packet Size
Continuous ARQ - Type I

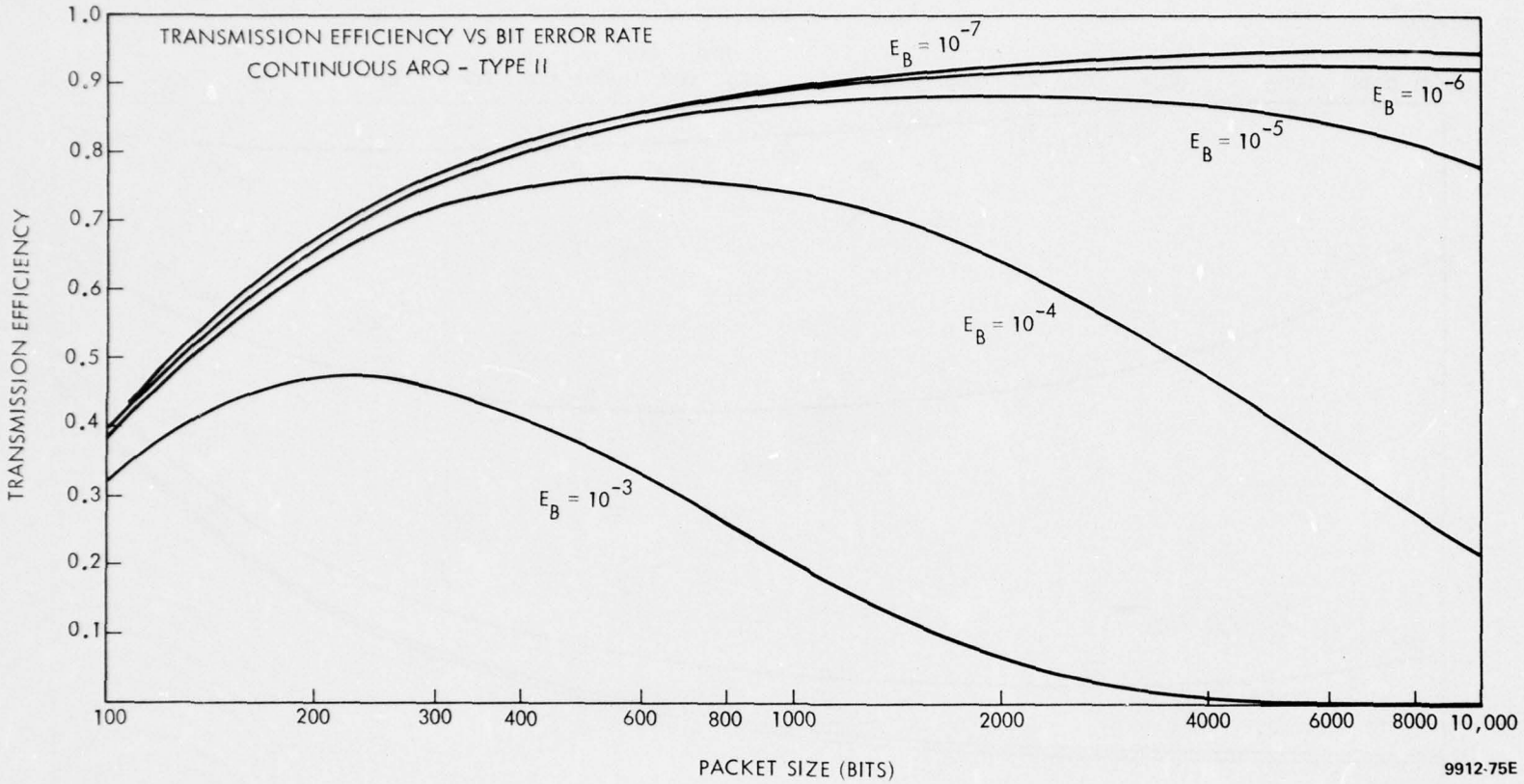


Figure 10-14. Transmission Efficiency vs Packet Size Continuous ARQ - Type II

10.2.4 Other CCIS Messages

Other CCIS messages are generated besides those used to set up Class I and II calls and AMA. These include maintenance messages, synchronization messages, and others summarized in Section 10.2.3.2.1 Future work would involve evaluating these messages to see if they contribute significantly to transmission overhead. Based on past experience, we feel that they will not contribute significantly and that the CCIS will remain at less than 1% of the total frame transmitted. When calculating the blocking probability of Class I calls and the queueing delays for Class II calls, we reserved about 1% of the frame for CCIS traffic. In light of the results obtained here, that estimate seems to be justified.

10.3 ANALYSIS OF BLOCKING AND DELAYS

10.3.1 Delays vs. Service

10.3.1.1 Problem

Probability of blocking of class I traffic and expected waiting time for class II traffic for peak traffic load conditions are prime performance parameters of the DAX. Whereas formulae for computation of these parameters under various input and service time distributions have been well documented, the method for computation in a flexible TDM link with adjustable boundary markers is not clear. In the conception and evaluation of performance of proposed DAX models it is necessary to develop computation procedures permitting easy computation and adequate accuracy for use in optimization of switching parameters such as the frame period, and decision on fixed or flexible boundary FTDM.

10.3.1.2 Objectives

Objectives of this task are:

a. to perform a literature search for procedures for computing data delay and class I blocking probability as a function of traffic load for a fixed and flexible boundary FTDM.

b. To evaluate computation techniques for accuracy and tractability.

c. To perform a parametric analysis of delay versus data load on a baseline DAX with typical trunk capacity, peak hour class I and class II load in terms of equivalent Erlangs.

d. To perform a preliminary investigation of the effect of noise on data load and delay.

10.3.1.3 Analysis and Results

10.3.1.3.1 Procedure

This section contains the initial results of an evaluation of blocking probability of class I traffic and the expected delay or response time of transmission of packet switched data and/or CCIS or other link/network control data for a typical multichannel intra DAX network trunk using FTDM as described by Coviello & Vena[19] and analyzed by Fischer and Harris[25].

The formulae used are the Erlang loss formula for voice or virtual circuit switched traffic (Class I) and the data delay approximation which is based on considering the data transmission as an M/M/N system rather than an M/D/N system.* A review of the derivations of these formulae showed the approximations introduce errors which are small compared to the frame interval especially when the channel utilization percentage is small. All errors are such as to produce conservative results. There is, however, a serious shortcoming of these classical formulae. It is the fact that they all compute the 'steady state' values of the expected loss probability and delay time assuming such values exist. In real telecommunication networks the traffic input parameters are time variable and there are periods of time when the average traffic load exceeds the available trunk capacities. During these periods the queue size and delay grow without limit in a delay system. If such overload conditions last for a short enough period they will leave the system intact but with a residual queue which must be dissolved before the steady state delay formula applies. This overflow condition could occur quite frequently in our data channels when extremely high trunk utilization factors exist. Kummerle [34] noted this overflow phenomenon and suggested a relatively crude correction term to be added to the classical steady state expected values of delay. Examination of his computed curves and Monte Carlo simulation results at several points show his procedure to under-estimate the delay and or queue size by factors as high as 2 to 1 when the data traffic load exceeds the value of trunk capacity exclusively reserved for data. Pending the development of procedures for more closely approximating the peak queues which will develop in a DAX system we will proceed to use the classical steady state estimates of delays and queue size. Care is recommended in estimating the buffer storage capacity required to prevent loss of data. High confidence levels or safety factors should be used and even then provisions for back-up overflow storage may be required.

---

* An M/M/N delay system is defined to be an N server system having a Markov input and exponentially distributed service times. The M/D/N system is an N server system with Markov input and fixed service time.

10.3.1.3.2 Numeric Results

As an illustrative example an analysis was made of a T_1 carrier used as an FTDM trunk carrying a combined estimated class 1 circuit switched load of 60 Erlangs and varying amounts of data in a packet switched mode. A frame interval of 10 milliseconds was assumed. An examination of the voice digitization procedures expected to be in use in 1985 indicated that the average data rate of a voice channel would be 15,500 bits per second or 155 bits per 10 millisecond frame. The 60 Erlang voice load was an order of magnitude estimate of the peak hour voice traffic on a typical interswitch trunk. The expected holding time per call was estimated as 5 minutes for voice traffic. Two trunk designs were evaluated.

In the first design 83 channels of the 100 in the T_1 trunk are allocated exclusively to voice. Each specific voice user would be allocated a slot width proportional to his digital rate so as to provide the required time transparency. The average data rate of 155 bits per frame is used as a normalized voice slot width. Data slots are considered and referred to as equivalent voice channels in stating the data traffic in Erlangs. An equivalent data channel consists of 155 bits per 10 millisecond frame or 15,500 bits/second. Data traffic may be carried in variable sized asynchronous packets. The 83 channels allocated to voice produce a probability of lost calls due to trunk blockage of less than 0.1 percent. Of the data traffic one equivalent channel was considered for CCIS traffic which is required for establishing and breaking down voice circuits and dynamic control of trunk channel allocation. The remaining 16 equivalent data channels would be used for packet switched data. The expected waiting times were computed for data loads varying from 1 to 15 equivalent Erlangs. The results of this fixed boundary trunk design were tabulated along with voice, data and total channel utilization factors.

The data waiting times and utilization factors were then computed for a trunk design using a flexible boundary for the voice region. Data loads up to 39 equivalent Erlangs were considered and the delays computed and tabulated. The results are shown in Table 10-14. Calculation of the values of the Erlang loss formulae were performed by hand calculator using Poisson approximation and then corrected, where necessary, for the finite state space and the non-integral value of the number of available channels (i.e., $\hat{s}$) for data.

TABLE 10-15. VARIATION OF WAITING TIME AND UTILIZATION WITH DATA TRAFFIC FOR FIXED AND VARIABLE BOUNDARY FDTM TRUNK

θ	$P_2=.01\theta$	PL 60 Erlangs of Voice & S = 83	EW/b N = 16 Fixed Boundary	EW/b N = 16 S = 83 Moving Boundary	Voice Utilization = P(1 - PL) P = E/S	Data Utilization Above Fixed Bound	Data Utilization Moving Boundary	Total Trunk Utilization $\frac{60(1-PL)+P_2}{99}$
100	1	.000857	1.50	1.50	.722720	.0325	.0256	.61564
200	2	.000857	1.50	1.50	.722720	.125	.051216	.62574
300	3	.000857	1.50	1.50	.722720	.1875	.076824	.635844
600	6	.000857	1.50005	1.50	.722720	.375	.153649	.666147
1000	10	.000857	1.5058	1.50	.722720	.625	.2560819	.70655
1500	15	.000857	2.2294	1.50	.722720	.9375	.384123	.757056
2000	20	.000857	∞	1.50	.722720	-	.5121638	.807561
2500	25	.000857	∞	1.500434	.722720	-	.640205	.858168
3000	30	.000857	∞	1.502	.722720	-	.7682458	.9085715
3500	35	.000857	∞	1.645118	.722720	-	.8962868	.9590765
3900	39	.000857	∞	21.525036	.722720	-	.9987196	.9994806

10.3.1.3.3 Interpretation of Results

A review of Table 10-14 produces the following generalized conclusions which should be used as guidelines in the evolution of the DAX detail design concept.

1) The data delays consist of a fixed component equal to 1.5 x the frame period + a variable additional delay due to expected input queue (t_w in the literature)

2) This additional delay has an expected value which is equal to

$$\bar{t}_w = \frac{B \cdot F}{N - E}$$

where

B = probability of a non zero input queue

N = channel capacity of trunk

E = Erlangs of input traffic

F = frame interval

3) Therefore the frame period should be kept as short as is feasible for retention of the dynamic allocation advantages with a given processor capacity.

4) The use of a T_1 carrier for a trunk carrying a busy hour traffic of 60 Erlangs of Voice traffic and 3 equivalent Erlangs of data would produce insignificant values of $\bar{t}_w$ even with fixed boundary trunk design.

5) The data traffic load will increase due to the need to ARQ in a noisy environment. This increase can be approximately calculated by the following formula*

$$E_{noise} = \frac{E_R\ (1 + OH)}{(1-E_P)\ (1-DC)} \approx \frac{E_R\ (1 + OH)}{1-DC}\ e^{NE_B}$$

where

E_{noise} = Erlangs of traffic in noise

E_R = Erlangs of traffic in error free environment

O_H = packeting overhead

DC = duty cycle of error burst

* More detailed mathematical models of the effect of noise on transmission efficiency are given in Section 10.2 and the optimum packet size versus the bit error rate is tabulated in that section.

E_p = packet error rate

E_B = bit error rate

N = number of bits in packet

$$E_p = 1 - (1-E_B)^N \simeq 1 - e^{-NB}$$

6) An evaluation of the increase in data traffic resulting from ARQ requirements assuming the noise environment specified for the TTC-39 showed that 3 Erlangs of data would grow to 7.827 Erlangs if a 620 bit data packet were used.

7) The load becomes 14.55 Erlangs under the same environment if the packet is doubled in size to 1240 bits.

8) Because of the transient delay error discussed by Kummerle, the accuracy of computed delays is not trustworthy in the region where the total trunk utilization is over 90% and the channels allocated exclusively for data have less capacity than the peak hour average data traffic requires. The problem of estimating peak momentary queue size or delay is accentuated by the use of FTDM. This is an inherent property of the technique and is the price that is paid for the increased trunk utilization. The effects on system design are the following:

a. Buffers should be designed for traffic load capacity satisfaction with high confidence levels to reduce frequency of overflow.

b. Back-up storage must be provided for overflow so as to prevent loss of data.

c. A minimum amount of trunk capacity must be allocated exclusively for data to prevent blocking or extraordinary delay of priority data traffic.

d. Blocking of low priority data traffic or ruthless pre-emption of voice traffic must be considered when buffer overflow is impending.

9) There is a third design concept of the trunk which would allow voice traffic to fill the entire trunk capacity rather than blocking after 83 channels in the trunk are busy. The parametric table results would still apply to that case. The only salient difference is that there will be an insignificant increase in total voice traffic carried (i.e. approximately .1 of an Erlang due to the decrease in blocking probability).

10) In Section 10.3.2, an evaluation of the delay of transmission of CCIS and other trunk control data over a trunk is made in order to estimate the time to set up and break down voice circuits. This data, along with that determined in Section 10.2, is useful towards estimating the required size and priority of the CCIS Class II data load.

10.3.2 Cross Office and Cross Network Delays

10.3.2.1 Problem

Cross network delay and response time are important performance parameters of any communication system. The tolerable limits are dependent upon the type of traffic handled. For instance a voice conversation will be perturbed if the cross-network delay in an established circuit exceeds more than a few hundred milliseconds. Data terminals communicating to remotely located computers in an interactive mode would like delays of less than a second or of less than the processing time of the computer if it is greater than a second. The cross-network delay of a communication network is dependent, among other factors, upon the network topology and operational procedures and the cross-office delay at each of the switching nodes in the circuit. The problem addressed in this section is the determination of the increase in cross-network delay caused by the need of the signal to pass through a termination or tandem switching node. This determination must be made for Class I (line type switching); Class II packet switching; Class II message switching; and CCIS and network control traffic.

10.3.2.2 Objectives

a. To clearly define the functions performed upon each of the above types of traffic at each node.

b. To derive mathematical relationships for relating delay to the parameters of the various switch elements performing the above mentioned functions.

c. To apply these relationships to the estimation of feasible values of cross-office delays to be expected using a base line switch and network concept with reasonable parametric variations.

d. To determine the critical parameters effecting cross-office delay and the sensitivity of delay to parameter variations.

e. To provide inputs for trade off studies wherein other performance parameters such as reliability, survivability, adaptability to change, and message security are considered along with delay to derive figure of merit and cost effectiveness relationships for system optimization.

f. To provide a basis for sizing of buffer storage and computer processing and core memory capacity.

g. To compute variation of probability of lost class I calls and Class II data delay as a function of traffic load.

h. To estimate the time to connect and break down a Class I circuit through the network.

i. To estimate the effect of noise environment on the expected data delay of Class II traffic.

10.3.2.3 Analysis and Results

The cross network delay of a typical DAX network is analyzed for five examples of traffic shown in Figure 10-15. These examples are:

a. A secure voice conversation transmitted as line switched Class I traffic at a rate of 32 KB/second.

b. An interactive data call between a remote terminal and a computer. The call is sent as packet switched Class II traffic using 64 KB/second as the loop rate of the computer and 2.4 KB/second as the loop rate of the remote terminal.

c. A FAX call sent as line switched Class I traffic at 2.4 KB/second.

d. A Class I/II teletype call between dislike terminals using message switching. TTY-1 is an asynchronous machine operating in Mode II at 45 baud in ITA-2. The recipient terminal is a 600 baud synchronous terminal operating in Mode I using ASCII code.

e. Trunk and Network Control Traffic: The class II section of each frame will be used for sending network and trunk monitoring and control messages such as CCIS, sync signals, TCCF network control and maintenance messages. As our example we will consider a CCIS message sent over a quasi-associative terrestial signalling circuit to set up a satellite link between DAX-1 and DAX-4.

10.3.2.3.1 Class I Secure Voice Conversation

From Figure 10-15 it can be seen that the circuit which would be established for this conversation would consist of DAX-1, DAX-2, DAX-5, two FTDM INTRA-NETWORK

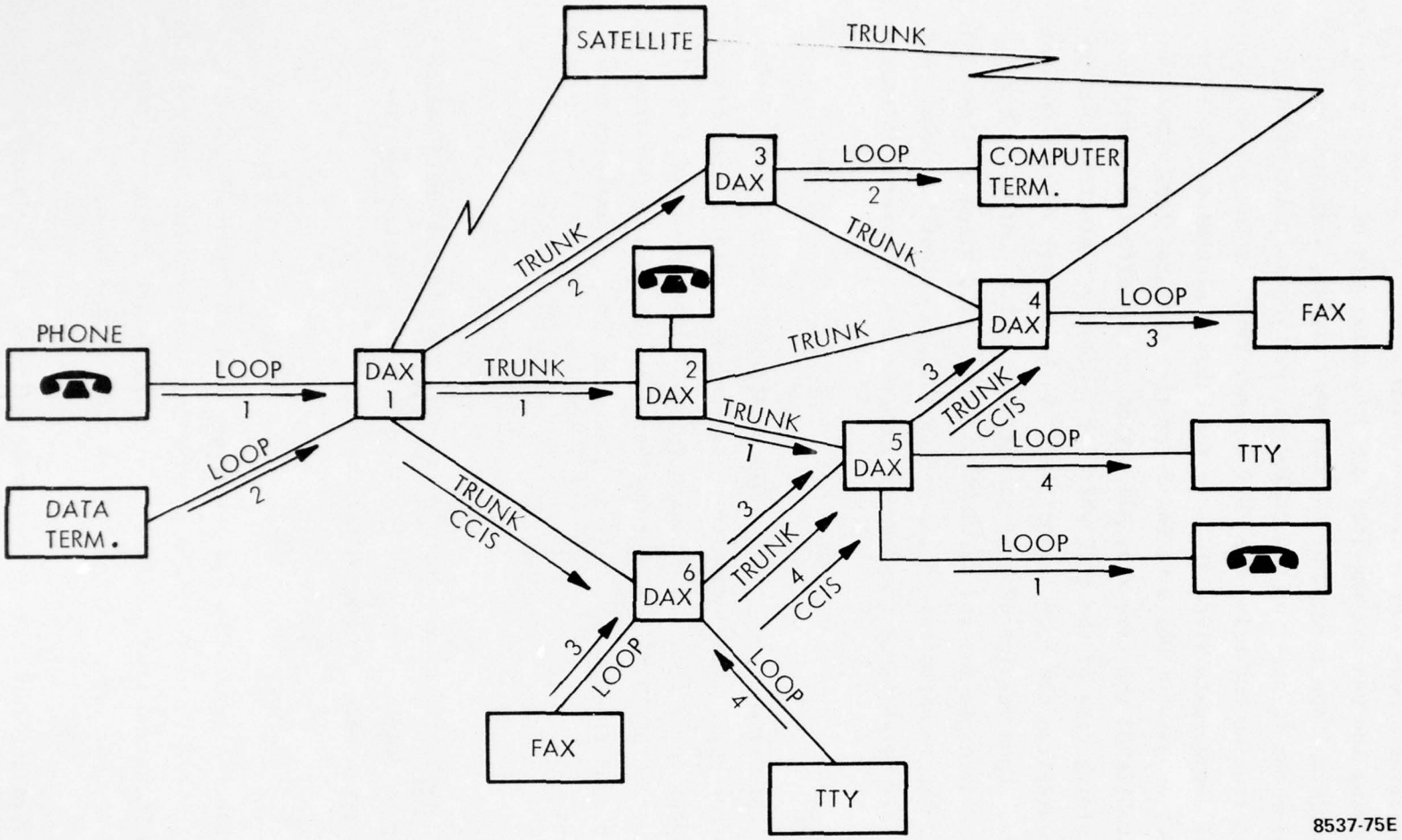


Figure 10-15. DAX Network with Typical Examples of Voice and Data Traffic

trunks and the two subscriber loops. As a Class I voice subscriber of the DAX, the output bit stream entering the DAX via the loop would be monitored by the scanner for an off-hook indication. Upon receipt of the off-hook signal, a digital receiver is assigned to determine the identity of the called party. The switch controller determines the desired routing and using the CCIS channel arranges for the reservation and allocation of appropriate width slots in the Class I section of the FTDM trunks connecting DAX-1 to DAX-2 & DAX-2 to DAX-5 for this conversation. Once the virtual circuit is established the time slot allocation on the effected trunks is irrevocable for the holding time of the call which is typically 5 minutes. During the course of the conversation the loop is monitored by the scanner for the on-hook digital signal pattern. Upon receipt of this pattern the switch controller again using the CCIS channels arranges for the termination of the allocation of time slots in the FTDM trunks and the recompacting of the residual Class I traffic slots. The response time of the call establishment and break down process is covered in example 5.

10.3.2.3.1.1 The Phone*

A typical digital secure voice phone would be the DSVT which uses 32 KB/S continuous variable slope delta (CVSD) modulation for analog to digital conversion. It includes a synchronous encryption section with a local clock which would be slaved to the DAX station clock. The 32 KB/S output digital signals would be converted to quasi-analog form by a conditioned diphase or dipulse modem for transmission over the loop to the switch.

10.3.2.3.1.2 The Loop

The loop would typically be a twisted pair such as WF-16 and would seldom be more than 4 kilometers in length. It would be 4-wire for full duplex operation. Transmission would be synchronous at the 32 KB/S rate.

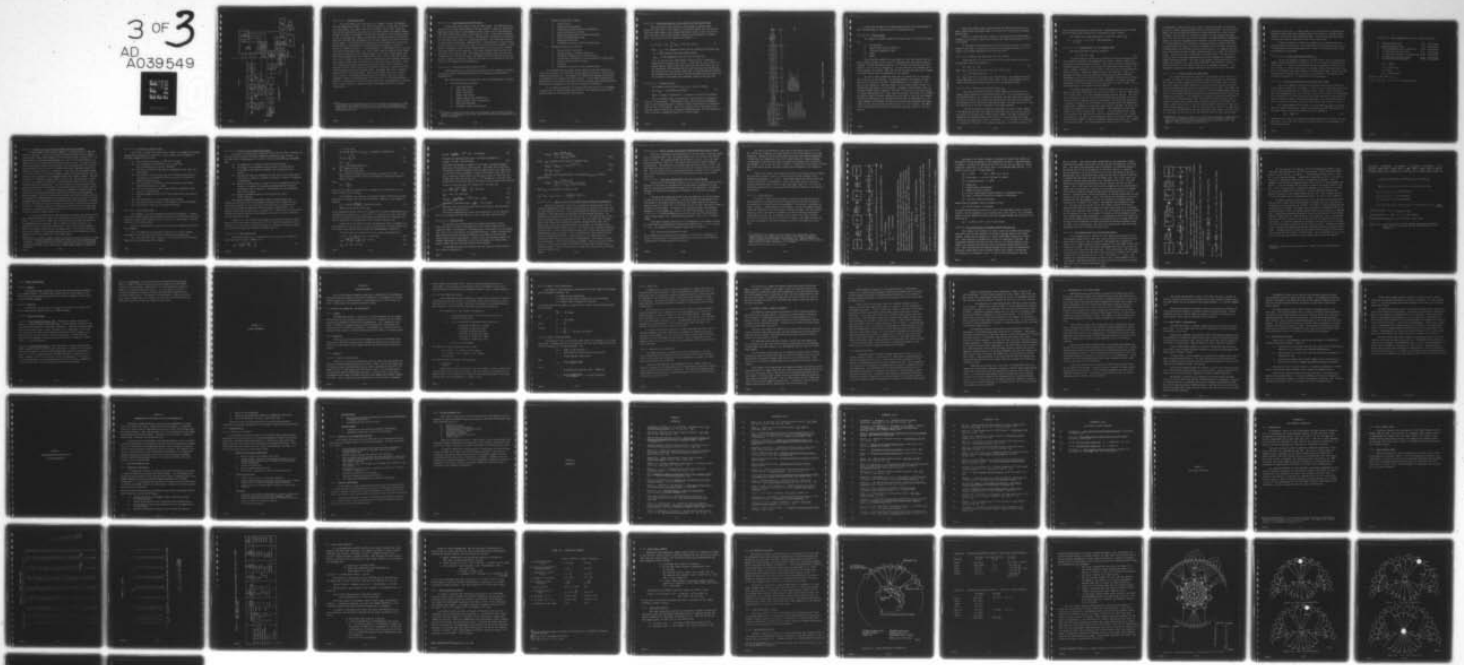
10.3.2.3.1 The DAX Switching Node

A functional block diagram of a DAX switch is shown in Figure 10-16. In the Class I voice connection DAX-1 and DAX-5 are termination nodes and DAX-2 is a tandem node. The subscriber loops and trunks interface with the switch through appropriate interface units which are part of the communication interface section.

* The functional descriptions for each of the elements in the example 1 of a voice connection apply to the other 4 examples and will not be repeated.

AD-A039 549 GTE SYLVANIA INC NEEDHAM HEIGHTS MASS ELECTRONIC SYS--ETC F/G 17/2
SENET-DAX STUDY. VOLUME 2.(U)
JUN 76
UNCLASSIFIED FR76-1-VOL-2 DCA100-75-C-0071
NL

3 OF 3
AD A039549



END

DATE FILMED
6-77

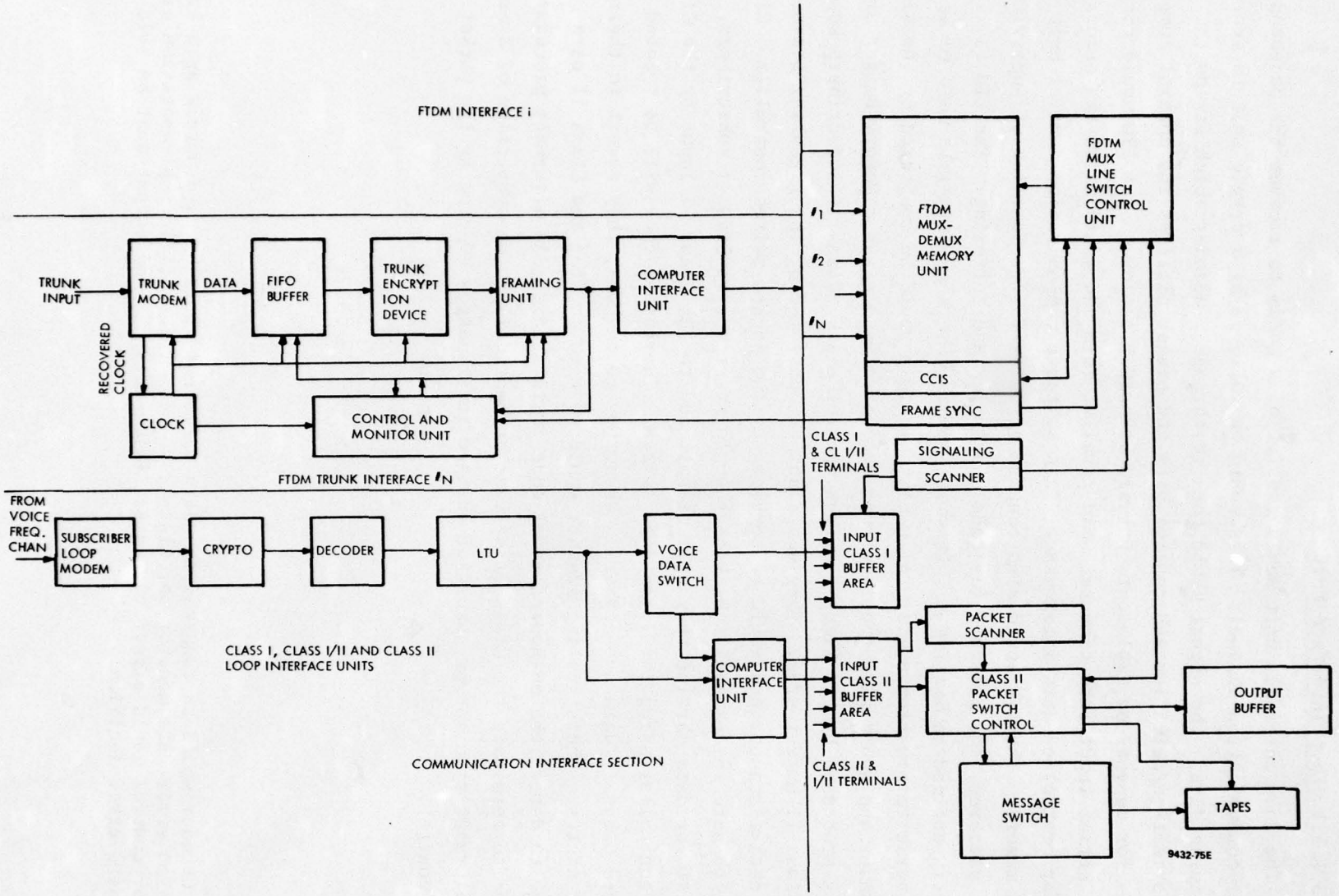


Figure 10-16. Functional Block Diagram of a DAX Switch

10.3.2.3.1.3.1 Loop Interface Unit

The loop interface unit would consist of a modem to recover the baseband from the communication channel. This would be followed by a crypto unit to decrypt the incoming data. The signal would then go through a decoder which strips the error correction/detection and control bits and either notifies the control computer of needs for repeats or implements Forward Error Correction.* The LTU would convert the bit serial input into computer word parallel form and would raise the necessary flags for controlled data interchange via a suitable communication control unit into the DAX memory. The LTU would also contain circuits for recognition of supervisory digital patterns, and possibly for code conversion and buffering. The LTU is more fully defined in Section 8. It would probably be a programmable unit to be used with diverse termination units. All Class I LTU's would be monitored by the signaling scanner and would have their outputs deposited in a Class I Memory Buffer area which is processed by the Class I switch control unit. Class II subscribers would use a Class II buffer area. They would not be monitored by the signaling scanner but by the packet scanner and would be serviced by the packet switch controller. Class I/II (alternate voice/data) subscribers would start out as Class I subscribers. If they dial for data service they will be switched to the Class II input by the Class I switch controller using the voice-data switch. Thereafter they will be treated as dedicated Class II data subscribers. When they go on-hook, they revert to their Class I status. The signaling scanner would have to monitor the Class II data exchange to detect the on-hook digital code in this case. An alternate procedure would be to terminate the Class II service automatically upon completion of a message after a preset time-out or number of message interchanges counted by the packet switch control.

* The decoder would be required for data calls but could be used merely as a transmission error rate monitor for Class I voice. The specific implementation of error control would effect the cross office delay if the output must be held up pending error decision.

10.3.2.3.1.3.2 Multichannel Trunk Interface Unit*

The DAX inter-node trunks will use the FTDM concept. The FTDM trunk interface unit will include those elements of the DAX required to terminate, decrypt and demultiplex these trunks for the purposes of drop, insert, switch, and monitoring individual channels when link-by-link encyrption is used. In the final implementation end-to-end encyrption will be used. The multiplex and buffering is done within the core memory under control of a designated processor. The trunk modem converts the trunk signal to base band. Synchronization is obtained either by phase locking of the trunk clock to the recovered clock or by the use of atomic standards. A FIFO buffer is included to allow for free running operation in the course of a limited time fade without loss of crypto sync. The trunk encryption device provides bulk encryption for the entire trunk. The framing unit maintains frame sync for start of frame and start of Class II region, the computer interface unit provides for bit serial to computer word parallel conversion and controls the transfers of data into and out of the computer on an interrupt basis.

10.3.2.3.1.3.3 Processing and Switch Control Section

The blocks shown in the processor and control section of Figure 10-16 depict the various functions performed and should not be construed to imply method of implementation.

The switching functions are broken down into the following types of service:

1. Line Switching which includes:
    a. Input loop control
    b. Input trunk control
    c. Output loop control
    d. Output trunk control
    e. Signal scanning and interpretation
    f. CCIS and control input interpretation
    g. Memory Map Maintenance
    h. Precedence control and processing

---

* Not shown on Figures 10-15 and 10-16 are Multichannel trunk interfaces to other networks using synchronous TDM or FDM. Such units would have to be provided if the service is required.

2. Packet Switching which includes:

   a. Packet scanner
   b. Packetizer and reassembler
   c. Transmission control character interpretation
   d. Input and output queue control
   e. Routing and output control character generation
   f. Packet switching between nodes
   g. Performance monitoring
   h. Precedence and security control and processing

3. Message switching which includes:

   a. Code and Format conversions
   b. Invalid message or procedure detection
   c. Storage for delayed and/or multiple delivery
   d. Full message accountability - log and journal or history tapes
   e. Message retrieval and/or recovery
   f. Intercept
   g. Precedence and security control and processing.

It is assumed that the message control characters from the terminal will indicate the services to be performed, the precedence, the security, as well as the addressee and message size. The computer will appropriately program the LTU based on information in a classmark and status register. It will monitor operation for malfunction on a real time or near real time basis to enable high system availability and minimize the probability of misrouted or compromised information.

It is further assumed that the computer speed and capacity will be adequate to ensure that all input and output control of all loops and trunks can be accomplished within the 10 millisecond frame interval.

10.3.2.3.1.4 Definitions Relative to Cross-office and Cross Network Delay

The cross network delay (TN) for a voice message is defined, using Figures 10-15 and 10-17, as the length of time from the instant that the first bit leaves the output buffer of the transmitting terminal (point A) until the time it leaves the input buffer of the recipient terminal to be reconverted to analogue voice. The value of Tn can be computed from formula 1.

$$T_n = 2\,T_{PL} + 2\,T_{SL} + \sum_{i=1}^{m} T_{ptr,i} + (n-1)\,T_{CT} + 2\,T_{ce} \qquad (1)$$

where: - T_{pL} is the propagation delays over the sending and receiving loop.

- T_{SL} is the sending and receiving terminal delays

- $T_{pTr,\ i}$ is the propagation delay over the i'th trunk and

-T_{CT} + T_{ce} are the cross office delays of each of the N-1 tandem and 2 terminal nodal switches in the circuit. In the case of voice circuits it is assumed that the terminal nodes have almost the same cross-office delay as the tandem nodes. This condition does not hold for packet and possibly message switching where the terminal nodes have to do packetizing and reassembly and accountability which may not be required of tandem nodes that merely control transfer of the packets and/or messages toward the recipient node.

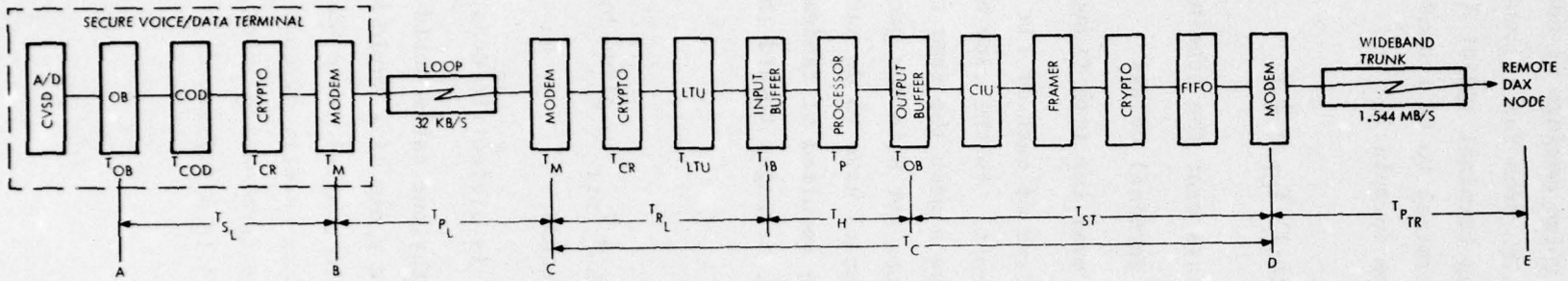
10.3.2.3.1.4.1 Propagation Delays

For unloaded cable the value of T_{PL} & T_{ptr} is given by formula

$$T_{PL} \text{ or } T_{ptr} = (\sqrt{\varepsilon})(L)/300 \text{ milliseconds} \qquad (2)$$

where L = length in Kilometers & ε is dielectric constant of the medium.

Thus a 4-kilometer loop with polyethylene cable would have a propagation delay of .02 milliseconds. A thousand mile radio link would have a propagation delay of 5.36 milliseconds while a 1000 mile coaxial cable would have a propagation delay of about .8 milliseconds. Thus a cross country circuit consisting of two 1500-mile coaxial links and two 4-kilometer terminating loops, all with polyethylene cable, would have a propagation delay of 24.18 milliseconds.



T_{SL} = PHONE TERMINAL SENDING TIME
T_{PL} = LOOP PROPAGATION TIME
T_{RL} = LINK RECEIVER TRANSFER TIME
T_H = HANDLING TIME
T_{ST} = TRUNK SENDING TRANSFER TIME
T_C = CROSS-OFFICE DELAY OF NODE
T_{PTR} = TRUNK PROPAGATION TIME
T_N = TOTAL CROSS NETWORK DELAY
T_E = SIGNAL EMISSION TIME (INCLUDED IN T_{SL} AND T_{ST})
T_P = PROCESSOR TIME
T_M = MODEM TIME

A/D CVSD = DSVT PHONE
OB = OUTPUT BUFFER
COD = ERROR CORRECTION OR SYNC CODING
CRYPTO = ENCRYPTION DEVICE
MODEM = MODULATOR/DEMODULATOR
LOOP = ASSUME A 32 KB LOOP
LTU = LINE TERMINATION UNIT
CIU = COMPUTER INTERFACE UNIT
FIFO = SPEED BUFFER
TRUNK = ASSUME 1.544 MBS T_1 TRUNK
IB = INPUT BUFFER

8536-75E

Figure 10-17. Signal Delay Time Diagram

To this must be added the 2 terminal delays and the cross office delays at the 3 switching nodes to obtain the entire cross-network delay.

10.3.2.3.1.4.2 Terminal Delays

In a digital voice/data terminal the terminal delay is the sum of the delays in:

a. A/D converter
b. Output buffer and Line Controller
c. Error and transmission coder
d. Crypto
e. Modem

The AD converter samples the voice signal at fixed time intervals and composes a digital message of one or more bits depending upon the method of digital encoding. For PCM the sampling interval is 125 useconds (8,000 cps) during which period a 6 to 8 bit word is transmitted. In Delta modulation the sampling rate is usually greater than 30 KB/S. When CVSD is used the digitizing rate can be dropped to 16 KB/S without excessive degradation of voice quality.

In class I voice transmission there must be a smooth flow of intelligent data in and out of the terminals. The minimum element of intelligence in PCM is the 6 or 8 bit word while in CVSD or Delta Modulation it is a single bit. The delay in the loading of a PCM output buffer is 125 μseconds whereas in CVSD, at a 32 KB/Second, this delay is 31.25 μseconds. Assuming a slaved terminal clock, no additional buffer delay is required. The coder, if used on voice to insert parity or crypto synchronization bits, would cause no additional delay since it is assumed that parity checks to monitor line quality will be done in real time with hardware and the loop signaling rate is adjusted to account for any additional bits. The crypto equipment also performs its function of pseudo noise modulation and demodulation in real time and hence also adds at most 1 or 2 bits of delay. The same is true of the modulator which emits an output band signaling element for each base band bit.

Define the terminal delay as the time from the instant the first bit of a signal element enters the output buffer till the last bit is emitted by the modem down the loop.

Then for a 32 KBS CVSD phone terminal the delay would be 219 microseconds assuming 2 bits of delay in each of the decoder, the crypto and the modem, and 1 bit in the buffer.

If the phone used 64 KBS PCM then the delay would also be 14/64 = 219 microseconds in this case. In both of these cases T_e, the message emission rate, is included in the sending termination of each loop and trunk.

10.3.2.3.1.4.3 Terminal and Tandem Node Cross-Office Delay

From Figure 10-17 it can be seen that the cross office delay of the input terminal mode is given by equation 3:

$$T_c = T_{RL} + T_h + T_{ST} \qquad (3)$$

Where $-T_{RL}$ is the delay in the receiving loop interface unit

$-T_h$ is the processing time

and $-T_{ST}$ is the delay in sending trunk interface unit. For tandem nodes the receiving interface is a trunk rather than a loop. In the output terminal node the trunk is the receiving interface and the loop is the sending interface.

10.3.2.3.1.4.3.1 Receiving Loop Interface Delay (T_{RL})

The delay in the receiving loop interface consists in the delay in the modem, crypto, decoder and line termination unit (LTU). As in the discussion of the termination delay it is assumed that the delays for these three units constitute the time for 6 bits at the assumed loop signaling rate since each of the functions are performed in real time while traversing the stage. The signaling delay of 1 bit for CVSD and 8 bits for PCM is covered in the sending interface.

The LTU is assumed to include an output buffer in which a signal element is assembled and held until the input line controller polls the unit and transfers the element to the input class I buffer area in memory. The input controller is assumed to have sufficient speed so that there is no queueing delay in the LTU (excluding all delays of less than a baud period - i.e. 31 μseconds for 32 KB/S and 15.5 μseconds for 64 KB/S). The average polling delay in the LTU is assumed to be

half of the baud period of the fastest loop. Assuming the fastest loop to be 200 KB/S, the delay in the LTU would be 2.5 useconds for class I traffic.

For a CVSD class I phone subscriber at 32 KB/S the value of T_{RL} =

$$6 \times \frac{1}{3.2 \times 10^4} + 2.5 \times 10^{-6} = 190 \text{ microseconds.}$$

10.3.2.3.1.4.3.2 Processor Delay (T_h) for Terminal Nodes

The processing delay is given by equation 4

$$T_h = T_{IB} + T_P + T_{OB} \qquad (4)$$

The major functions of the Class I processor are those associated with the establishment, monitoring, and breaking down of virtual circuits in addition to administrative and maintenance functions. These latter operations are performed as background programs and hence do not affect the delay time of voice traffic. The Class I traffic is handled as a loss system so that there can be no queueing due to traffic overloads. Instead the surplus calls are lost (blocked). There is no operation on traffic other than input and output processing and the possible monitoring of error rates to detect failures. Terminals involved in any call must be entirely compatible and transmission time transparent. The input and output processing will be fast enough so that the only delays for Class I traffic will be the polling delays in the input and output buffers.

The input loop processor will transfer an element of (R)(F) bits per frame period to the input buffer where R is the baud rate and F is the trunk frame period. If R = 32 KB/S and F = .01, the information element would be 320 bits. The information should not be placed in an output trunk frame until a complete 320 bit element is received. Upon the receipt of the 320th bit the output trunk controller is enabled to output the 320 bit element in its allocated slot on the appropriate output trunk. Assuming a T_1 trunk with a data rate of 1.544 MB/S, the entire 320 bits must be output in 207 μseconds starting at some random point of the class I region of the 10 millisecond frame. The position in the frame will also change in the course of the conversation as calls are completed and the class I region is recompacted. The delay in the outputting of the first bit on the output trunk after the trunk output is enabled is completely random varying from 0 to a complete 10 millisecond frame. This average additional delay will be 1/2 a frame or

5 milliseconds. Thus the total value of T_h at a node will average 15 milliseconds for Class I traffic varying uniformly from 10 to 20 milliseconds. The Class I input buffer area for our postulated 32 KB/S voice subscriber must be at least 640 bits in length* since the loop will continue to output bits after enabling the output and no bits can be lost or sync will be destroyed. The buffers must have additional capacity to allow for propagation delay variations and clock drifts during fades. It is rather arbitrary as to whether we consider the delay to be part of the input or output buffer since there is actually no movement of data in memory and the processing functions are performed during the waiting period. For compatibility with class II data delays we will consider 10 milliseconds to be T_p and 6 milliseconds to be T_{OB}. T_{iB} for voice will be considered to be essentially 0. For data class II calls T_{iB} will be the normal queuing delays which are dependent on instantaneous channel utilization factors. The value of T_{OB} includes 5 milliseconds for the randomness of the output slot in the frame and an added 1 millisecond to allow for variations of up to 35 bits during a fade or due to propagation anomalies. It is implemented by delaying the output trunk enable until 352 bits are in the buffer.

10.3.2.3.1.4.3.3 Processor Delay for Tandem Nodes

In a tandem node, where the input for a voice channel comes in bursts in a T_1 trunk, and leaves in bursts in an output T_1 trunk the situation is quite different. Since the input transmission rate is the same as the output rate, it may be possible to enable the output controller as soon as the first word has been transferred into memory without any fear of overwriting or loss of data. The buffer area for our 32 KBS channel need only to be 320 bits. The total handling delay would consist of 5 milliseconds T_{OB} and the processing time would consist of the time for the computer to I/O 640 bits. For the processor to handle I/O trunks this would have to be less than 42 microseconds per trunk. The problem with this concept is that it would be necessary for the output trunk controller to ensure that an output voice channel, which started out in a time slot coming slightly later than its input position, and hence having a T_{OB} of close to zero, did not move ahead of its input due to more rapid call terminations on the input trunk than on the output. Were this event to occur during the course of a call, the same data would be read twice and hence

* An additional input buffer capacity of 320 bits may be required at tandem nodes to prevent loss of synchronization since the position of the channel in both the input and output frames can vary independently in the course of a call. (See Section 4 - Network synchronization and control.)

crypto sync would be lost. A safer concept is to allow 640 bits of storage and to require an output trunk enable dependent upon the existence of a complete frame in storage as in the case of the terminating nodes. The 10 milliseconds processing time (T_p) would also be applicable in the case of tandem nodes.

In order to maintain a conservative estimate of cross-network delay the second concept will be utilized. This would mean that the expected value of T_h for a tandem node is 15 milliseconds made up of a value of 10 milliseconds for T_p and T_{OB} averaging 5 milliseconds.

10.3.2.3.1.4.3.4 Output Trunk Interface Delay (T_{ST})

T_{ST} for this example is defined as the time from the first bit entering the computer Interface unit until the 320th bit of our voice channel is transmitted out on the trunk. It includes 207 microseconds for the transmission of 320 bits over the 1.544 MB/S T_1 trunk plus the delays in the Modem, crypto, FIFO and Framer. The FIFO is not needed in the output channel of the link and the modem, crypto and framer can be assumed to require 6 bits of delay or 4 microseconds. The value T_{ST} for a 32 KB/S voice channel in a T_1 trunk would be 211 microseconds.

10.3.2.3.1.5 Total Cross Network Delay for 32 Kbps Voice Circuit

Table 10-16 summarizes the total cross network delay for the 32 KBS voice circuit shown in Figure 10-15 as example 1. The total cross network delay after the circuit has been established is 72.79 milliseconds. Of this 24.18 milliseconds is due to the propagation delay over a 3000 mile circuit in coax cable. Of the remainder 38.4 milliseconds is the cross office delay for 3 DAX nodes and .41 milliseconds is due to the terminal delays. The critical parameters in the cross office delays of Class I traffic are the frame interval and the terminal loop transmission rates.

The random component of the delay is the half frame T_{OB} delay at each node. Since this delay distribution is assumed to be uniform over the frame period (F) the total variance of the cross network is given by equation 5:

$$\sigma^2_{TN} = \frac{(n+1)}{12} F^2 \qquad (5)$$

Assuming that the total T_N is normally distributed around into 72.79 means the maximum class I voice delay over a 3000 mile circuit would be 79.29 milliseconds at a 90% confidence level.

TABLE 10-16. CROSS NETWORK DELAY FOR 32 Kb/s CLASS I VOICE CALL

1.	Sending Terminal delay	0.22	milliseconds
2.	Receiving Terminal delay	0.19	milliseconds
3.	^{2X}loop propagation delay (4 KM cables)	0.04	milliseconds
4.	2 x 1500 mile coax trunk links	24.14	milliseconds
5.	2 x cross office delays for terminal nodes	32.80	milliseconds
6.	1 Tandem node cross office delay	15.40	milliseconds
	T_N = Total cross network delay	72.79	milliseconds

$$\sigma_{TN}^2 = \frac{n+1}{12} F^2$$

F = 10 milliseconds

n = 2 links

σ_{TN} = 5 milliseconds

Class I T_N, 90% = $\overline{TN} + 1.3\ \sigma_{TN}$

Class I = T_N, 90% = 72.79 + 6.5 = 79.29 milliseconds

10.3.2.3.2 Interactive Data Call Between Remote Terminal and Computer

In this example assume that the computer is a dedicated class II data subscriber to the DAX. As such, its LTU would interface directly with the computer interface unit rather than through the voice data switch shown in Figure 10-16. The remote terminal in contrast would probably be a class I/II subscriber which is allocated a port on its data switch by dialing for data service. If no port is available it would enter a queue to be allocated a port in accordance with its priority. When obtaining its port, it will, via suitable subscriber signaling, identify the computer to which it wishes to be connected. A CCIS like call origination message will be sent to DAX #3 to which the computer is homed to determine whether the line controller of the computer can accommodate another active remote terminal. When the affirmation is attained, an enable signal is sent to the remote terminal to notify it that it can commence to send data. It will send idle characters till it has a message prepared for transmission. Its LTU will filter all idle characters except possibly the first and will forward all non-idle message bits to the input class II buffer area of DAX No. 1. The packet scanner will examine the messager header and control bits to determine the priority and enter the message in the appropriate processing queue. The DAX #1 packet switch processor will check for errors and send the appropriate ACK/NAK message to the terminal and will enter the packet in the appropriate queue for transmission over the FIDM link to DAX #3 upon which the computer is homed. The packet will not be released at DAX #1 until an ACK has been received from DAX #3. The packet need not be transmitted across the link in a single frame but once its transmission has begun no other class II bits can be sent across that link until the end of packet flag is sent.*

In DAX #3 the packet will be removed from the link under the direction of its input trunk controller and stored in the input buffer area of its class II packet switch. The DAX #3 packet scanner will read the priority and place the packet in the appropriate processing queue. When it reaches the top of the queue it is checked for errors. If none are found an Ack is sent to DAX #1 and the packet is placed in the output buffer area for the computer loop. The processor will keep track of all packets in a multipacket message and will not start outputting to the computer until all the packets are in the output buffer.

*This concept is extremely important since it considers the Class II region of a FTDM trunk as a dedicated asynchronous communication channel from one DAX packet switch to another. It allows optimization of frame size and packet size to be performed essentially independently for bit serial to computer word parallel conversion and controls the transfer of data into and out of the computer memory on a DMA basis instigated.

10.3.2.3.2.1 Illustrative Parameter Values

In the case of interactive traffic from a terminal to a computer the average message will be between 600 to 6000 bits. For the purpose of this example the following assumptions will be made:

(1) The terminal query message is 3000 bits in length

(2) The Link between DAX-1 and DAX-3 is a T_1 link which has a transmission rate (R) of 1.544 M B/S.

(3) The link is 1500 miles long and has a propagation delay (T_{Ptr}) of 12 milliseconds.

(4) The link has a bit error rate (E_B) of 10^{-3} except during bursts of .05 duty cycle (DC) during which the error rate is so high as to render the trunk useless.

(5) The noise bursts last from 2½ to 50 milliseconds and are spaced from 50 milliseconds to 1 second.

(6) The two loops are each 4 KM in length and have propagation delays (T_{PL}) of .02 milliseconds

(7) The loop random error rate is 10^{-4} with a burst duty cycle equal to .005.

(8) The frame interval is 10 milliseconds on all trunks.

(9) The peak hour voice load (E_I) is 80 Erlangs at 15,440 bits/channel.

(10) The peak hour load (E_{II}) is 4 equivalent Erlangs.

10.3.2.3.2.2 Cross Network Delay

As in example 1, the cross network delay is given by equation 1. However in the case of data messages the delay must be defined to last from the instant the first bit is sent by the transmitter until the last bit of the message is received correctly at the output terminal.

There is a twofold effect of data errors upon cross network delay that has to be computed:

(1) The number and size of packets that have to be sent to complete a message have to be increased to allow for overhead and retransmissions.

(2) The total data traffic load has to be increased and the result upon trunk utilization and queuing delays computed.

10.3.2.3.2.3 Packet Size and Transmission Overhead

In Section 10.2 transmission overhead was defined and the effect of packet size and error control procedure on Packet transmission efficiency (E_{ff}) analyzed. It was shown that an optimum packet size exists in terms of maximizing E_{ff} and depends upon:

(1) the probability of a packet being received with errors;

(2) the number of non-information bits per packet (c) which have to be included in the packet for message, link, network and error control

(3) the time (W) after the transmission of the last bit of a packet until a decision is reached as to whether the packet is correct, or requires retransmission;

(4) whether the ARQ procedure is block by block or continuous and if continuous as to whether all transmissions commencing with the block for which the NAK is received is retransmitted or only the specific block that is NAK ed;

(5) the data transmission rate R of the link and the class I traffic Load (E_I).

Using typical baseline DAX parameter values, optimized packet size, transmission efficiency and other performance parameters were computed for varying values of channel bit error rate (E_B) for Block by Block, continuous with total retransmission and continuous with single packet retransmission. The results are tabulated in Tables 10-9, 10-10 and 10-11. Tables 10-12, 10-13 and 10-14 tabulate transmission efficiency with variation of packet size and bit error rate for the three types of ARQ.

Figures 10-12, 10-13 and 10-14 are graphs of transmission efficiency versus packet size for error rates varying from 10^{-3} to 10^{-7} and the three forms of ARQ. Figures 10-9, 10-10 and 10-11 show the variation of transmission efficiency versus bit error rate for each type of ARQ and varying packet size.

10.3.2.3.2.3.1 Block by Block ARQ

The optimum number (I_{opt}) of information bits in a packet in a block by block ARQ system is given by Equation (6).

$$I_{opt} = \frac{K}{2}\left(\sqrt{1 + \frac{4}{KE_B}} - 1\right) \qquad (6)$$

$$K = C + WR\,(1-DC) \tag{7}$$

The optimum efficiency ($E_{ff,opt}$) is computed in equation (8):

$$E_{ff,opt} = \frac{I\ell^{-PE_B}}{I_{opt} + K} \tag{8}$$

$$P_{opt} = I_{opt} + C \tag{9}$$

P_{opt} is the optimum packet size

and E_B = random bit error rate

DC = error burst duty cycle.

R_{II} = effective transmission rate for class II data on the trunk. It is equal to the trunk rate R times the percentage of capacity not used for class I traffic.

Hence

$$R_{II} = \frac{CH-E_I}{CH} \tag{10}$$

where CH is the number of equivalent voice channels in the trunk and E_I is the class I traffic load in Erlangs.

In our example a T_1 trunk has an equivalent capacity of 100 voice circuits digitized at an average rate of 15,440 bits/second. Hence R_{II} in our example is given by:

$$R_{II} = 1{,}544{,}000 \left(\frac{100-80}{100}\right) = 308{,}800 \text{ bits/second} \tag{11}$$

during the busy voice traffic hour.

In the current DAX concept, data will be packetized in variable length packets up to a maximum size to optimize E_{ff}. The value of C is taken to be 60 bits since there are 40 to 48 control bits in the ADCCP format and additional non-information bits will be required in the information field for security, priority, etc.

In the case of our 1500 mile trunk, the propagation delay is assumed to be 12 milliseconds in each direction. Assume therefore that W = 30 millisecond total. This allows 6 milliseconds for processing and acknowledgement transmission.

The total value of K during the busy boice hour would therefore be:

$$K = 60 + (.03)\ (308{,}800)\ (.95) = 8860.8 \text{ bits} \tag{12}$$

$$I_{opt} = \frac{8860}{2}\left(\sqrt{1 + \frac{4}{8.860}} - 1\right) = 907 \text{ bits} \tag{13}$$

$$P_{opt} = 60 + 907 = 967 \text{ bits} \tag{14}$$

$$E_{ff,opt} = \frac{907}{8860+907} \ \ell^{-.967} \ (.95) = 3.35 \text{ percent} \quad (15)$$

The peak hour equivalent data load in the noise environment is:

$$E_{II}/E_{ff} = 4/.0335 = 119.26 \text{ Erlangs.} \quad (16)$$

Since this load exceeds the capacity of the trunk an overflow condition would exist resulting in an infinite queue. This means that it would not be feasible to carry 4 Erlangs of class II traffic across a T_1 trunk in the .001 bit error noise environment if block by block ARQ is used. The value of E_{II} would have to drop below 0.67 equivalent Eralngs or 10.345 bits per second to get below the overflow limit. At 3 messages per second the utilization factor of the trunk would be 87 percent of the 20 channel data capacity.

If the bit error rate were 10^{-4} instead of 10^{-3} we would have

$$I_{opt} = \frac{8860}{2} \left(\sqrt{1 + \frac{4}{.8860}} \ -1 \right) = 5973 \text{ bits} \quad (17)$$

$$P_{opt} = 5973 + 60 = 6033 \text{ bits} \quad (18)$$

$$E_{ff,opt} = \frac{5973}{8860 + 5973} \ \ell^{-.6033} \ (1\text{-DC}) = 2.092\% \quad (19)$$

$$\text{The class II load under noise} = \frac{4}{.2092} = 19.12 \text{ Erlangs} \quad (20)$$

The trunk class II peak utilization factor would then be .956 which would be an unsafe operating level.

The results computed above indicate the inadvisability, if not impracticability, of using Block by Block ARQ over a noisy, wide-band, 1500 mile trunk.

10.3.2.3.2.3.2 Continuous ARQ

The two cases of continuous ARQ to be considered are where all data after the receipt of a NAK is retransmitted or on the contrary where only the single NAK block is retransmitted. Continuous ARQ is only possible on a full duplex link such as will be available in the proposed DAK network. Either continuous ARQ mode produces far better transmission efficiency than Block by Block but both require greater data storage capacity. Repeat of only the NAK packet produces the greatest improvement in efficiency in the noisy environments but requires both greater storage capacity and a more complex control program.

The optimum efficiency for continuous ARQ with repeat of all data after a NAK is given by equation 21a:

$$E_{ff,opt} = \frac{I_{opt}\, \ell^{-P_{opt}E_B}\, (1-DC)}{I + K - WR(1-DC)\ell^{-P_{opt}E_B}} \qquad (21a)$$

where I_{opt} is given by the solution of equation (22a):

$$E_B I_{opt}^2 + KE_B I_{opt} - K + (K-C)\ell^{-P_{opt}E_B} = 0 \qquad (22a)$$

$$\text{and } P_{opt} = I_{opt} + C \qquad (23)$$

When only the incorrect packet is retransmitted $E_{ff,opt}$ is given by equation (21b):

$$E_{ff,opt} = \frac{I_{opt}\, \ell^{-P_{opt}E_B}\, (1-DC)}{I_{opt} + C + P_{opt}\, (1-DC)(1-\ell^{-P_{opt}E_B})} \qquad (21b)$$

where I_{opt} is the solution of equation (22b)

$$I_{opt}^2\, (E_B) + I_{opt}\, (C)(E_B) - C + C\left(\frac{1-DC}{2-DC}\right)\ell^{-P_{opt}E_B} = 0 \qquad (22b)$$

10.3.2.3.2.3.2.1 Numeric Results for Continuous ARQ with Complete Retransmission

Substituting parameter values for our example the optimum packet size for continuous ARQ operation with complete retransmission was found to be 352 bits with the .001 trunk bit error rate. The optimum transmission efficiency was 6.58% which represents a 100% improvement over the block by block optimum efficiency. The probability of a packet received with errors was 29.7%. The peak data load in the noise environment was 60.76 equivalent Erlangs. This far exceeds the 20 Erlangs which is the nominal data transmission capacity available during a busy voice channel hour. Hence this procedure would not be feasible for 4 Erlangs of data traffic during a peak 80 Erlang voice load. From Table 10-10 (Transmission Overhead of Section 10.2), 1.185 Erlangs would be the peak Class II data load that could be handled during a busy voice hour. This is postulated on a 90% utilization factor for data including CCIS traffic. In Section 10.3.1 it was shown that a 90% utilization factor for data would result in approximately a 10% increase in delay to input queueing. This would be perfectly acceptable. During a slow voice traffic hour (i.e. a voice traffic load of 20 Erlangs), the peak Class II traffic which would be handled is 1.30 Erlangs hence the 4 Erlangs load would will not be feasible with this method of ARQ.

10.3.2.3.2.3.2.2 Numeric Results for Continuous ARQ Repeating Only Errored Packet

Table 10-11 (transmission Overhead-Section 10.2) shows that if only the errored packet is repeated after a NAK, the optimum packet size in a .001 bit error environment is 224 bits. The optimum transmission efficiency is 46.7%. This is 14 times bettern than block by block and 7 times better than continuous ARQ with complete retransmission. The probability of a packet being received in error is 20.07%. The peak trunk utilization factor is 42.75 of the link data capacity during a busy voice hour. From the results of the trunk queueing analysis in Section 10.3.1 the queueing delay on the trunk would be less than 0.1 milliseconds per packet due to traffic congestion.

10.3.2.3.2.3.2.3 Total Cross Network Delay for Class II Query Message

Based on the results given in Section 10.3.2.3.2.3.2.2 we will assume the data packets to be variable to a maximum of 227 bits per packet including the 60 overhead bits and that continuous ARQ with repeat of only the NAK packet. Our 3000 bit message will therefore be transmitted as 18 packets, all except the last having 227 bits.

The sum of the propagation delays on the two loops will be .04 milliseconds and that of the trunk 12 milliseconds. As far as the delay for appearance of the first packet at the output from its emission at the input terminal, the same procedure is used as in Section 10.3.2.3.1 except that now the basic information element is all the bits in the packet. The values T_e in T_{SL} and T_{ST} become the predominant parameters. In our example T_{SL} = 94.58 + .19 = 94.77 millisecond. T_{ST} for DAX-1 = .735 + 0.4 = 1.135 milliseconds since the 227 bits are sent over the available class II trunk capacity (i.e. 308.8K B/S) T_{ST} for DAX-3 = 3.55 + .19 = 3.74 milliseconds.

The number of packets which have to be transmitted at the terminal and each of the DAX units are determined by the Pascal distribution with r = 18, q = E_p and P = 1-E_p. The expected value of incorrect packets ($\overline{N_e}$) is given by

$\overline{N_e}$ = rq/p+r and the variance σ by rq/p^2.

For the SENET trunk an average of 22.587 packets will have to be transmitted to obtain 18 correct packets. 26 packets would have to be sent to be more than 90% confident that 18 correct packs will be correct.

For each of the termination loops with the postulated error rate of 10^{-4}, $\overline{N}_e$ = 18.4132 and 19 packets will ensure delivery of 18 correct packets at over 90% confidence. Because of the fact that packets can transit the network independently of each other, the computation of the cross network delay for a complete message is complicated by the fact that there is overlap in the delays of the various circuit elements. By the time the remote terminal is transmitting the last packet to DAX-1, many of the earlier packets would have already been transferred to the computer by DAX-3*.

Figure 10-18 presents the assumptions relevent to processing time and computes the cross network delay as equal to 2.522 seconds at the 90% confidence level. Lengthening the network (i.e. 3 links - total length 3000 miles) would increase the result by only about 45 milliseconds. If the cross network delay is defined to exclude transmission delays across the input and output loops (i.e. as is done in the TTC-39 specification) the cross network delay would be reduced to less than 100 milliseconds.

10.3.2.3.3 Class I FAX Call

This case is very similar to example 1 except that the transmission overhead would have to include provisions for forward error correction since ARQ is not possible as it would destroy time transparency,** an inviolate requirement for class I traffic. A second difference from example 1 is that the minimum essential element of information is the full content of a page of output. The cross network delay (T_N) in this case must be considered to be the time from the beginning of transmission of a page till the completion of the page at the output terminal. This is the sum of the time for the first bit to arrive at the output terminal and the time to transfer all the bits describing a picture at the modem rate of the loops.

---

* It is assumed in this example that the terminal will form packets and the computer perform the operations of reassembly of the message, stripping of control sections and acknowledging and accountability. Normally terminal DAX units will perform these functions. The receiving DAX would then have to receive packets before starting transmission to its terminal.

**Assuming buffered terminals are not used.

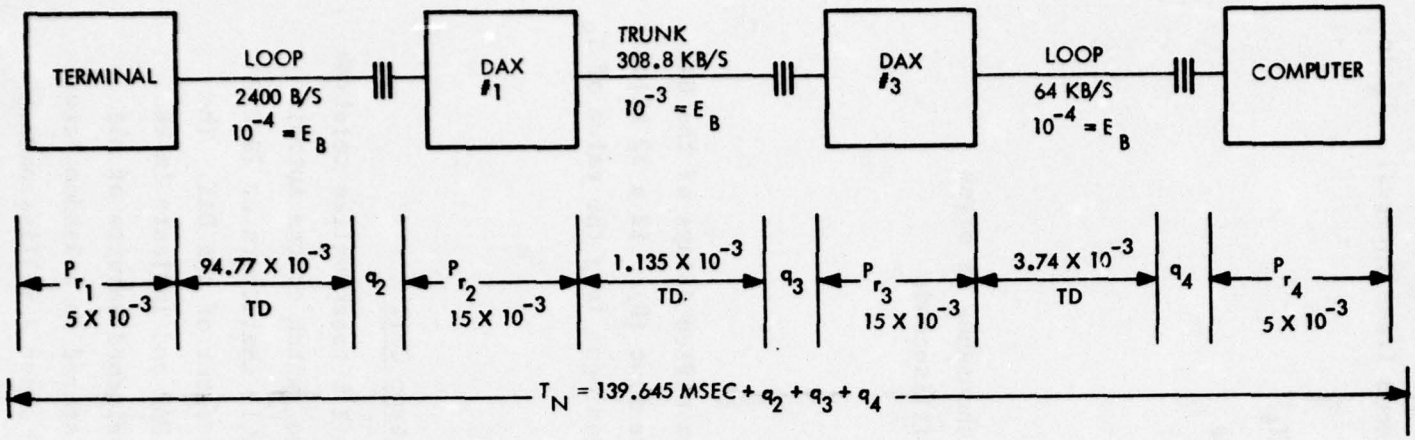


Figure 10-18. Cross Network Delay for Packet-Switched (Class II) (Example 2) 9616-75E

assume that P_{r1} = 5 milliseconds

$P_{r2} = P_{r3}$ = 15 milliseconds

P_{r4} = 5 milliseconds

These times actually are the buffer intervals rather than the processing times. For processing assume each processor to have capacity of 1,000,000 instructions/second and that this enables it to process 100 messages/second. The time to process a packet is taken as 1 millisecond (i.e., 10 packets per message). Assume that σ_{Pr} = .3 milliseconds and that processor utilization factor is 70% under our peak load. Using Pollaczek-Khintchin's formula for input queue waiting time we get

$$T_{qi} = \frac{1 \times .7}{2\ (1-.7)}\ (1.09) = 1.27 \text{ milliseconds per queue}$$

The cross network delay time for the entire message is the time to send 25 packets across the input loop (largest transmission delay (TD) + the time to send the last packet across the entire network.

$T_N = 25 \times 94.67 + (5 + 94.77 + 1.27 + 15 + 1.135 + 1.27 + 15 + 3.74 + 1.27 + 5 + 12.4) = 2.52$ seconds

at the 90% confidence level when the expected value of cross network delay is 2.2 seconds.

According to our traffic estimates a FAX page will require the transfer of 200 kilobits of data (including 5:1 data compression) in class I, Using the results of example 1, the delay for transfer of the first bit (T_F) will be taken as 25 milliseconds for a 3000 mile, 2 link circuit with a 10 millisecond frame interval. T_N in seconds is given by the following equation:

$$T_N = 2 \times 10^5/R + .016F\ (N+1) + .009M = T_E + .001\ T_F$$

T_N = cross Network delay for 1 FAX page in seconds

T_E = Transmission delay = $2 \times 10^5/R$

R = Modem rate

F = Frame Period in milliseconds

N = Number of links in circuit

M = line of sight distance between terminals in thousands of miles

T_F = Delay is receipt of first bit at output in milliseconds

T_F = 1.6F (N+1) +9M in milliseconds.

Substituting the parameter values of example 3 we get

T_N = 83.408 seconds = 1.39 minutes

Less than 50 milliseconds of the total is due to the cross-office delays of the DAX switches. The key parameter in T_N is the data rate of the line (R). If a 32 kilobit digital loop were substituted for the 2400 band assumed analogue loop the value of T_N would drop to:

$$T_N = 2 \times 10^5/3.2 \times 10^4 + .075) = 6.325 \text{ seconds.}$$

10.3.2.3.4 Cross-network Delay for Message Switched Teletype Call

This example is an instance of a class II data call between unlike teletype subscribers such as would currently use the TTC-39 message switch or the Autodin network. Assume that the message has a length of 2500 ASCII characters or 20,000 bits. Each subscriber is assumed to be a class I/II subscriber of the DAX. The transmitting terminal would dial for data service to its DAX and indicate in the header of his message that he desires store and forward switched service of his call. This would mean that the complete message will be stored in a Random Access Peripheral storage device with an average latency time of about 8 milliseconds. The message will be handled as a unit. It will be fully processed at each node

that it transits. This entails header interpretation, code conversion to ASCII validation of precedence and security routing, and full accountability in history tapes. The input loop transmission rate is equivalent to 6 characters/second or 416 seconds till the end of message will be received at DAX #6. It would be grossly uneconomical to tie up 20,000 bits of core for that length of time for a message whose end to end delivery delay cannot be less than 11 minutes at best due to transmission time across the input and output loops. This is the reason core resident in-transit packet switching is not indicated for this case. The cross-office delay for message switching using RAS for intransit will be less than 2 seconds. This is the value that will be used in computing the cross-network delay. Assume that the processor has a capacity of 16 messages per second or an average service time of .0625 per message and that under our peak data load it is 70% utilized. Assuming Poisson arrival distribution, the average queuing delay at each processor will be .080 seconds. Figure 10-19 depicts the computation of the cross-network delay from the beginning of first bit transmitted to complete message received to be 11 minutes 27.61 seconds. Excluding the loop transmission times (i.e., the delay from the receipt of last bit of EOM to receipt by the output terminal of the first bit of the output) the cross network delay is 4.277 seconds. If the circuit included 3 links instead of 1, the cross-network delay would increase by 4.29 seconds or a total of 11 minutes 31.9 seconds with loop transmission and 8.567 seconds without loop transmission. If the message was split into packets and sent by core resident packet switching the processing time of a message would drop to about 10 milliseconds and the output link would be transmitting packets while the input is still transmitting. The total cross network delay would then become essentially the time to traverse the slowest link.

10.3.2.3.5 Cross Network Delay for a CCIS Message Sequence

In the example we will consider the cross network delay incurred by the CCIS message sequence which is sent over a quasi-associative terrestrial path and used to set up a Class I call on a satellite link between DAX-1 and DAX-4 of Figure 10-15. Assume a 5000 mile circuit composed of the three SENET trunks, each 1000 miles in length. Table 10-16 gives the propagation delay of this circuit as 24.14 msec (or 8.04 msec per 1000 mile link). Cross office delays and input queue waiting time are taken to be the same as those specified in example 2 for a packet switched call (see Figure 10-18); i.e., Pr_i = 15 ms and Tq_i = 1.27 msec. The emission for an average CCIS message (50 bits) on the T_1 trunk computes to be 0.1 milliseconds. With these definitions the delay incurred in setting up this particular call can be calculated (see Figure 10-20).

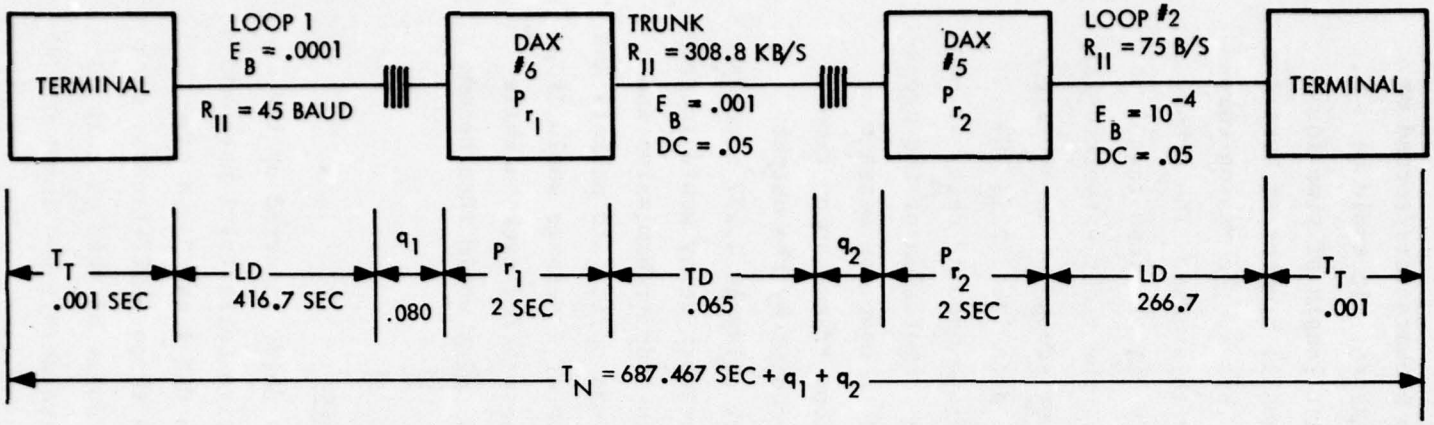


Figure 10-19. Message Switched TTY (Dislike Terminals) (Example 4) 9517-75E

Assume that the processor has a capacity of 1,000,000 instructions/second which enables it to process 16 messages/second or an average service time $\overline{S}$ = .0625

Assume during peak hour data traffic the peak utilization factor of the processor is 70%. Then using Pollaczek-Khintchin's formula

$$T_q = \frac{.0625\ (.7)}{2\ (1-.7)} \left(1 + \left(\frac{.019}{.0625}\right)^2\right) = .080 \text{ seconds.}$$ This is the queueing time on q_1 & q_2.

$$T_N = \frac{7.5 \times 2500}{45} + \frac{20{,}000}{75} \times 2 \times .001 + 2 \times .08 + 2 \times 2 + .065 + .051 = 11 \text{ minutes } 27.61 \text{ seconds}$$

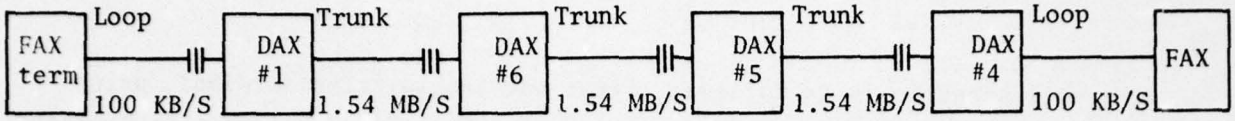
where .051 is the estimate of time for buffering and propagation delays.

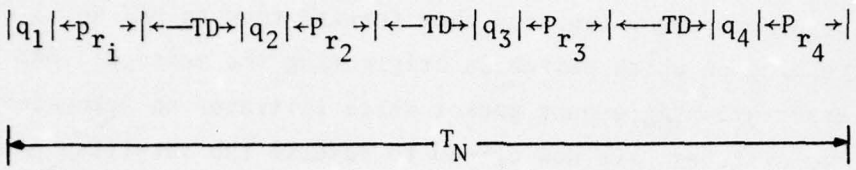
The call scenario is as follows: First the transmitting terminal, using subscriber signaling identifies the receiving terminal. Then a Reservation Request packet is sent by DAX No. 1 to DAX No. 4. The intermediate DAX's, DAX No. 6 and No. 5, do not process messages. Instead they forward them to DAX No. 1 or Dax No. 4, respectively, depending on which switch is originating the message. DAX No. 4 responds with a Reservation Agreement packet which initiates an Acknowledgment from DAX No. 1. The two switches have now agreed to reserve the satellite path for the call. While the called party is being rung, Ringback is sent to DAX No. 1 so that it can provide ringback to its subscriber. When the subscriber answers, DAX-4 requests allocation of the channel reserved for the call. DAX-1 responds with an Allocation Agreement which is acknowledged by DAX-4. The call is now in progress.

The delay incurred in setting up the call is the subscriber signaling time* plus the time it takes for each CCIS message to traverse the circuit plus the time waiting for the called party to go off-hook. The latter is subscriber dependent and is estimated as 5 seconds. The frames can be estimated by cumulating the delay of each CCIS message as the call scenario progresses until path connection is completed. Thus 7 single packet messages multiplied by the cross network delay of each message yields an overall delay of .6265 seconds for call setup time. This is considerably less than the time it would be to set up the same call by using the satellite link exclusively. It should be noted that no portion of the satellite link is allocated until both subscribers are off hook and are ready to converse. Transmission capacity is only allocated when absolutely necessary, providing efficient utilization of the trunk. Including subscriber signaling and recipient alerting time the illustrative FAX call could be established in 12.6265 seconds. After establishment of crypto sync which should take much less than a second, information transfer can commence.

---

*Subscriber signaling time is estimated as 7 seconds for multitone push button terminals.





$P_{r_1} = P_{r_2} = P_{r_3} = P_{r_4}$ = 15 milliseconds

$q_1 = q_2 = q_3 = q_4$ = 1.27 milliseconds

TD = 8.047 + 0.1 = 8.147 milliseconds

T_N = 1.27 + 15 + 8.147 + 1.27 + 15 + 8.147 + 1.27 + 15 + 8.147 + 1.27 + 15 = .0895 seconds

Call Set up Time = 7 x T_n + $T_{signaling}$ + T_{alert}

Call Set up Time = 7 x .0895 + 7 + 5 = 12.6265 seconds

DAX network contribution to call set up time = .6265 seconds

Figure 10-20. Cross Network Delay for CCIS messages needed for establishing a Class I circuit via a satellite link for high-speed FAX call. (Example 5)

10.3.3 Crypto Induced Delays

10.3.3.1 Problem

Cross network delay and response time are important performance parameters of a switched network. The normal cross network and cross office delays in the DAX system have been analyzed in Section 10.3.2 with the exception of those delays caused by COMSEC equipment.

10.3.3.2 Objective

The objective of this task is to assess the increase in cross office and cross network delays caused by the use of COMSEC equipment.

10.3.3.3 Analysis and Results

10.3.3.3.1 Trunk Encryption Devices (TED) - When used to provide traffic flow security on trunks between DAX switching nodes, the TED's operate continuously and are essentially transparent to individual subscribers. Since they operate synchronously at the trunk bit rate, a delay of several bits is negligible. Loss of TED sync causes loss of traffic on all channels until the DAX switch recognizes loss of master frame sync and initiates TED resync procedures. The TED resync may require several thousand bit intervals (at the trunk rate) in addition to several round trip transmission delays.

10.3.3.3.2 Loop Encryption Devices - The stream cipher technique used on modern COMSEC equipment does not introduce significant delays in transmission once crypto sync has been achieved. The call signaling procedures necessary to establish a secure call require signalling between the COMSEC equipment in addition to the normal exchange. Since this time occurs only at the beginning of a call and is essentially equivalent to the usual waiting intervals for dial tone, ringback and off-hook signalling, it does not interfere with subscriber operation.

10.3.3.3.3 Conclusions - We conclude from the foregoing that no significant amount is added to cross-office and cross-network delays by the used COMSEC terminal or trunk equipment. Maintenance of bit synchronization on lines and trunks is a necessary prerequisite to this. In later studies, it will be necessary to determine the precise interface and operational characteristics of selected COMSEC devices, and evaluate the system impact. However, we believe that, whatever complexity may be added to the system by the use of these devices additional significant transmission delays will not occur.

SECTION 11

SYSTEM ASSESSMENTS

SECTION 11

SYSTEM ASSESSMENTS

This section contains a general evaluation of capabilities of the SENET-DAX switching system with respect to system size, modularity, and expandibility, traffic-handling capability, system availability, interoperability with other systems, security considerations, and system and technical control.

11.1 SYSTEM SIZE, MODULARITY, AND EXPANDABILITY

11.1.1 Problem

The SENET-DAX concept is a new and as yet untried approach for the integrated processing of voice and data, consisting of a set of techniques that look promising from the standpoints of feasibility and performance. It is important at this point to examine the conceptual design of the DAX architecture to determine what limits may exist on system size, modularity, and expandability, because of the importance of these factors in eventual realization of the concept in a working system.

11.1.2 Objective

This section will be based on the recommended software and hardware architecture of the SENET-DAX concept, and will examine the question of size and growth capabilities with regard to these aspects and the calculated traffic-handling capability.

11.1.3 Analysis

11.1.3.1 System Size Considerations

The size of a conventional switching system has always been described by defining the minimum and maximum quantities of lines, trunks, and service terminations, with grade-of-service specifications based on traffic generated per subscriber, the number and size of trunk groups and the blocking, if any, in the switching matrix. The DAX differs from the conventional systems in a variety of ways. The system accommodates three classes of traffic with each class composed of different transmission rates. Such a system can be more appropriately sized in terms of throughput

than by number of subscriber lines. Since the trunk transmission media is T_1 (1.544 Mb/s), the transmission rate provides a unit of throughput and the corresponding system (size) generating the throughput provides a unit of system expansion.

11.1.3.2 Unit Subscriber System

A Unit Subscriber System is defined as a system entity serving voice and data subscribers such that all non-local (line/trunk) traffic can be served by a single link equivalent to one T_1 carrier. Given this, the number of voice subscribers and data subscribers served by the Unit Subscriber System can be determined.

The assumptions for voice traffic are (Appendix):

a. An average holding time of 3 minutes for local calls and 5 minutes for trunk calls

b. A call distribution by transmission rate is given by:

10 percent at 2400 b/sec (vocoder)
10 percent at 4000 b/sec (LPC)
15 percent at 8000 b/sec (APC)
50 percent at 16,000 b/sec (CVSD)
10 percent at 32,000 b/sec (CVSD)
5 percent at 50,000 b/sec (KY3)

c. All calls are full-duplex.

The effective weighted transmission rate is then

$$(0.1 \times 2400) + (0.1 \times 4000) + (0.15 \times 8000)$$
$$+ (0.5 \times 16000) + (0.1 \times 32000) + (.05 \times 50000)$$
$$= 15,440 \text{ b/sec}$$

The maximum number of available servers becomes

$$\frac{1.544 \times 10^6}{15,440} = 100$$

Assume that 83 servers are used to carry voice traffic, 1 server carries CCIS messages and 16 servers carry data traffic. Then 60 erlangs of voice traffic with a grade-of-service of 0.1 percent and 3 erlangs of equivalent data traffic with insignificant delay can be accomodated on a single T_1 link.

11.1.3.3 Number of Voice Subscribers

The number of voice subscribers that generate the trunk traffic of 60 erlangs can be determined as follows. Let

N = number of voice subscribers

A = total two-way traffic (erlangs) for the N subscribers

a = two-way traffic (erlangs) per subscriber

If we assume that 2/3 of traffic is carried over the trunk (line-to-trunk and trunk-to-line), then

$$\frac{2A}{3} = 60 \text{ erlangs}$$

and

$$A = 90 \text{ erlangs.}$$

Since $N = \frac{A}{a}$

letting $a = 0.3$,

$$N = \frac{90}{0.3} = 300 \text{ voice subscribers}$$

11.1.3.4 Number of Data Subscribers

From Section 11.1.3.2, we have data trunk traffic of 3 erlangs, or 3 x 15,440 bits/second. Assuming that this is two-thirds of the total data traffic on the link, that total becomes 4.5 x 15,440 bits/second. Let

N_1 = Number of data terminals

λ = total number of two-way message per busy hour

ℓ = average message length (bits)

Then

$$N_1 = \frac{(4.5 \times 15{,}440)\ (3600)}{\lambda \ell}$$

Given

λ = 24 messages per busy hour, and ℓ = 28000 bits

$$N_1 = \frac{(4.5 \times 15{,}440)\ (3600)}{(24)\ (28000)} = 372 \text{ data subscribers}$$

11.1.3.5 Switch Size

It appears that a single T_1 link has the capacity to support 300 voice subscribers and 372 data subscribers. Since the minimum size of the system is one T_1 trunk, where the T_1 link has been defined as having the ability to carry local traffic generated by Unit Subscriber System, a single switch can economically handle 672 subscribers. The system architecture is totally flexible, however, to allow a lesser subscriber population and higher connectivity of (partially used) T_1 trunks to other nodes.

The maximum size of a DAX can be up to 16 T_1 trunks in a purely tandem exchange configuration or any combination of lines and trunks up to a total of 16 lines and trunks to handle any mixture of local and tandem traffic. This upper bound on maximum size of 16 T_1 trunks is based on a cycle time for solid state memory of 215 nanoseconds (Section 8.3.2.3). Although larger systems sizes based on availability of faster memories will be possible in the 1980 time frame, it is doubtful if larger sizes will be required in the DCS.

It should be noted that although the system architecture has been based on a moveable boundary, the above calculations for purposes of simplification have assumed a fixed boundary. Also, the effect of increase in data traffic resulting from ARQ in a noisy environment has been ignored. The two assumptions have an opposite effect on the Class II traffic carrying capability, but do not necessarily cancel each other.

11.1.3.6 Expandability and Modularity

The expansion concept is based on two levels of modularity. The first level of modularity allows expansions on per line basis for the local subscriber switch. The second level of modularity allows expansion of T_1 trunks (or equivalent links for subscriber expansion).

Overall system modularity and expandability can be readily achieved by emphasis on modularity and expandability in the subsystem-level functional elements, such as the processor and software architecture. That functional modularity and growth potential are inherent in system architecture can be seen from the descriptions given in Sections 7 and 8.

The selection of a highly distributed microprocessor-based architecture makes it possible to consider a very modular system with expansion optimized to considerations of transmission media and accommodation to new subscriber services. This system approach covers a wide range of sizes for DCS switching centrals without the need to resort to inefficient approaches for large system sizes, such as have been configured in the past by interconnecting smaller systems and using interconnecting trunks.

11.2 ASSESSMENT OF TRAFFIC HANDLING CAPABILITY

The SENET-DAX system provides more effective traffic handling capability than conventional switching systems. Utilization of trunk groups is between 50 and 70 percent in a conventional fixed channel size system engineered for .01 grade of service. Hence, approximately 40 percent of costly transmission capacity is unused in the conventional system. A network using the SENET switching concept greatly increases trunk utilization.

Data and voice traffic are combined in the same switched network, resulting in more efficient trunk groups than when used for separate real time switched and store and forward networks. The trunk pools provide greater trunk utilization factors for a given grade of service.

The SENET-DAX system adjustable channel sizes provide trunk channel bit rates tailored to the data rate, thus eliminating inefficiencies of multisampling on fixed higher rate data channels; e.g., multisampling 2400 b/s data on a 16 Kb/s trunk channel.

In the network, trunk groups can be designed with higher overall blocking probabilities while maintaining the same real time voice channel grade of service. Data packets are inserted in empty trunk slots on an as-available basis. Data packets can, if desired, be delayed and give first priority to the voice (Class I) traffic.

Data packets of the same message are adaptively routed over different routes, spreading them over a larger trunk pool with resulting increased trunking efficiency. Also, full duplex data calls in the network are not assigned a dedicated full duplex channel. Therefore, data packets of other switched messages can occupy the return path of the duplex data call. This in essence places two full duplex data calls on an equivalent single full duplex path. A conventional circuit switched network would require two full duplex paths, one for each data call.

Traffic handling capability of the SENET-DAX switch, as described in Section 11.1, provides switch throughput capability much larger than the estimated 1985 average switch traffic (originating and terminating). Even further expansion of matrix throughput is possible by varying the First-In-First-Out buffer and data bus sizes.

Traffic data for the network (Table A-5, Appendix) indicate Class II (delayed or store and forward) data traffic is 13 percent of total originating and terminating traffic. Class II tandem data traffic is expected to be a larger percentage of total tandem traffic because data packets having lower priority than Class I (voice/real time) traffic will encounter longer network paths (more trunk links). Each additional trunk link in excess of the number used by voice calls represents an increase in network data traffic. Error control redundancy will also increase Class II traffic relative to Class I traffic. Contrasting with these increases, Class II data traffic is decreased, as stated previously, by the fact that data packets from different calls use opposite sides (transmit and receive) of a full duplex channel. In view of these considerations, the estimated 1985 Class II data trunk traffic is expected to be in the area of 10 to 30 percent of total trunk traffic. This quantity of Class II data trunk traffic can occupy the expected excess trunking capacity while providing full service to Class I (voice/real time) traffic, and thus achieve a significant increase in trunk utilization and resultant transmission system cost reduction.

11.3 SYSTEM AVAILABILITY

A switching system availability commonly specified is .9999 (e.g., the TRI-TAC AN/TTC-39 switch). An availability of .9999 results in approximately one hour of system down time per year. A less costly system availability of .999 results in approximately 10 hours of switch downtime per year. The acceptability of a given switch availability is dependent upon its effect on communication network performance. The distribution of the resulting downtime and the cost of providing increased system availability are factors to be considered in specifying switch availability. If downtime occurs mainly as widely distributed failures with a failure affecting only a small percentage of total switch capability, a lower availability might be acceptable if it results in significant cost saving.

Switch availability must also be considered in light of overall network availability. If other network components, such as radio, cable plant, etc. have failure rates significantly higher than the switch, the advantages accruing from increased switch availability, obtained at some cost, will not be realized.

The definition of system downtime which determines system availability is complex. Downtime for a specific number of failed communication terminals is generally accrued at a rate based upon the ratio of failed terminals to total terminals. Thresholds of percent capacity lost are often set to establish when a given system failure is defined as a total system failure. In the AN/TTC-39 specification, failure of more than 20 percent of the terminals is considered a total system failure and downtime is accrued at the full rate. If failure affects 20 percent or fewer terminations, downtime is weighted according to the percentage of termination capacity lost.

The SENET-DAX switching system provides high availability by means of a combination of distributed processing control and unit redundancy. Input-output processors are distributed so that they interface a small percentage of the switch's line/trunk capacity. An individual unit failure thus affects only a small fraction of switch capability, i.e., much less than 20 percent, enhancing overall system availability. Automatic replacement of the input-output processing modules is one method by which further improvement in system availability can be achieved. The common equipments, which include the Nodal processor, Routing and Overflow Memories, and Nodal Clock can be provided redundantly with main and standby units. When a unit failure is detected, the standby units are automatically switched into service. Reducing downtime to the period when both main and standby units have failed provides the level of system availability demanded by switched communication systems.

Subsystems which branch out to many units of the SENET-DAX switch, for example, the switch timing subsystem, originate at duplexed common control equipments, such as the nodal clock, and then branch to numerous units within the switch. As the switch timing spreads farther throughout the switch, smaller fractions of the switch capability are dependent upon the timing distribution circuits. At a selected level in the timing distribution hierarchy, transition from reliance on system redundancy to distributed control can be made to provide cost-effective switch availability.

11.4 INTEROPERABILITY WITH OTHER SYSTEMS

SENET-DAX switching system ability to adjust trunk channel size to the size of the information channel to be switched not only makes more effective use of transmission capacity, but also provides a switching system which is compatible with a wide range of existing and future communication terminals and systems. Inventory equipments within the United States and foreign countries already use many different data bit rates. The size of any slot in the SENET multiplex structure can be adjusted to provide a channel bit rate equal to any of these. In fact, the switch provides a universal trunk channel the size of which can be tailored to any desired channel bit rate. For example, a 16 Kb/s channel uses a 20-character multiplex slot while a 48-Kb/s channel uses a 60-character slot. slot.

Future digital systems, many of which have not yet reached the concept stage, will appear and demand new and different switched communication service. A breakthrough in voice coding technology could radically reduce the channel bit rate required for digital voice communications. The SENET-DAX switching system capability for flexible dynamic allocation on trunk channels can readily service these yet unknown data bit rates through adjustable slots in its multiplex structure.

The 10-millisec buffer storage inherent in switching system design, and the fact that the system provides buffers on an individual channel basis, permits operation with terminals and networks which have independent timing and relaxed timing standard accuracies. If bit integrity mist be maintained for 1 day, timing standard frequency differences in the order of 10^{-7} are permissable. If bit integrity need only be maintained for 1 hour, larger standard frequency differencies in the area of 10^{-6} are permitted.

Electrical interfaces can be expected to differ among different interfacing terminals and networks. Interface units are necessary to adjust between the electrical characteristics of the interfacing equipments and those of the SENET-DAX switching system. The interface units may also provide any special signaling control interfaces that are required.

The stored program control, chiefly in the nodal processor, provides the flexible signaling system which can operate with the many different signaling/routing plans used by the various domestic and foreign networks, e.g., TRI-TAC, EUROCOM, etc.

Stored program control also permits selection of different algorithms that can perform routing according to different routing and numbering plans. Gateway interfaces can also be readily added via software to provide for connections to other networks having their own numbering and routing plans.

11.5 SPEECH SECURITY CONSIDERATIONS

The SENET-DAX system is a digital communication system for both voice and data traffic. It thus lends itself to the use of encryption to provide communication security to its users.

Digital loop signaling using repeated digital code signaling techinques can be encrypted by the subscriber terminal crypto unit to provide signaling and traffic flow security for subscriber loops. Pools of loop crypto units can be provided at the switch to decrypt/encrypt signaling between it and its subscriber terminals.

Digital trunk groups maintain bit integrity through buffering and thereby will permit circuit-switched subscriber terminals to use end-to-end encryption when crypto Stage II becomes a reality.

In Crypto Stage I, link encryption is provided on the digital trunk links by means of trunk encryption devices. The trunk encryption devices bulk-encrypt the links in order to secure calls which are not end-to-end encrypted, as well as trunk signaling and framing for traffic flow security.

Switch stored program control and software can control the crypto unit pools and key distribution systems provided at a switching node. Software routines in the control can also provide traffic segregation for special COMSEC communities.

The analysis in Section 10.3.3 indicates that network delays introduced in the call path by both trunk and loop encryption devices are negligible. Increase in signaling time due to subscriber loop security equipment coordination and additional trunk signaling data for COMSEC purposes is incurred only at call initiation, and does not appreciably change the usual waiting times that include dial tone, ringback and off-hook signaling.

The SENET-DAX system can use loop and trunk encryption devices presently being developed for the tri-service military networks. The fundamental encryption process is unchanged in the SENET-DAX network and the methods of applying the various loop and trunk encryption devices are similar to those used in the TRI-TAC digital network.

The Trunk Encryption Device (TED) is used for encrypting high bit rate trunk links. The Loop Key Generator (LKG) is used at the switch to encrypt/decrypt the supervision and signaling from secure voice terminals and the data (including addressing) messages from secure data terminals. The Dedicated Loop Encryption Device (DLED) is used to provide encryption/decryption at the secure data subscriber terminal. The TRI-TAC Digital Secure Voice Terminal (DSVT) incorporates a Loop Key Generator (LKG) within the terminal itself. The Automatic Key Distribution System (AKDC) can be controlled by the SENET-DAX central processor, and may provide efficient distribution of keys to the terminals and switches of the network.

11.6 SYSTEM/TECHNICAL CONTROL

Three levels of System/Technical Control are envisioned for the SENET-DAX switched communication network. These are:

a. System Control that provides near real time control of the communication network, as well as broad system planning and engineering

b. Nodal Control that provides near real time control and management of a communication node

c. System Control Support that provides support data to System Control, executes system control commands and performs equipment maintenance.

The major objective of System Control is to optimize performance of a network encountering changing traffic demands, network damage and equipment failures.

Nodal Control allocates circuits to switches and users in accordance with direction from System Control. It assesses performance of circuits and equipment under its jurisdiction, performs, or refers to subordinate activities, fault isolation and required corrective action, and reports node status to System Control.

System Control Support performs equipment and channel quality control monitoring and equipment maintenance at a specific equipment, e.g., the switching center. It reports equipment and channel performance and traffic data to System and/or Nodal Control.

The SENET-DAX switching system can perform the System Control Support and a portion of the Nodal Control functions. The switch can perform fault detection by means of both hardware and software built-in test equipment. Switch maintenance would be accomplished by replacement of a card or other least replaceable unit. Frame channel errors counted by built-in test equipment are reported to the nodal or system control facilities as a measure of link performance. Traffic data on trunk links is accumulated in the control processor and reported to the System Control facility for use in planning network modifications that will improve network performance. Numbering plan and routing plan update commands from the System Control Facility are received by each switch, which executes update of its data base.

The nodal control functions of channel and group patching are divided between the switch and the nodal control facility. The channel patching function is inherent in the operation of the SENET-DAX switch, with its flexible multiplex slot format, and is, therefore, performed by the switch itself. The group patching function which provides drop and insert of trunk groups at a node is performed at the Nodal Control Facility. T_1 or other size DAX trunk groups are multiplexed into super groups for transmission via radio and cable plant at the Nodal Control Facility.

SECTION 12

RECOMMENDATIONS FOR FURTHER STUDY AND EXPERIMENTATION

SECTION 12

RECOMMENDATIONS FOR FURTHER STUDY AND EXPERIMENTATION

Although the SENET-DAX Study, to date, has been comprehensive in scope, areas still remain that require further definition and analysis. In addition, there is an obvious need for experimentation with the concepts and techniques that have been postulated. These investigations can in turn be divided into what appear to be near-term and far-term efforts. "Near-term" in this context is intended to mean by 1980. "Far-term" lies beyond that point.

The basic objective of further study and experimentation will be to obtain a better understanding of the feasibility and behavior of the techniques considered thus far for application to the SENET-DAX concept. Other techniques evolving in today's rapidly developing technology must also be evaluated. The tangible result will be a better definition of the architectural requirements for the SENET-DAX approach, especially with regard to first-level access switching in strategic military networks handling large volumes of voice and data traffic with a wide variety of bit rates and formats.

12.1 NEAR-TERM RECOMMENDATIONS

12.1.1 Definition and Analysis

It is important to define a realistic functional and electrical environment in which the SENET-DAX architecture may someday operate. This environment would include the volume and mix of voice and data access traffic, random and burst noise, COMSEC equipment utilization, satellite network characteristics, and other considerations. A close interaction with the DCA is required for these definitions, so that the environment at which the architecture is aimed will most nearly approximate that which is predicted.

Based upon these definitions and the techniques being studied, profitable areas of analysis appear to include the following:

a. Continuing analysis of throughput, delays, blocking, and other performance measures

b. Optimal design objectives for switch and network synchronization

c. Varying trunk bit rates versus performance, size, and complexity of the architecture

d. Software and hardware techniques for key variable distribution

e. Satellite link performance

f. Impact of rate/code/format conversion on SENET-DAX architecture

g. Use of FEC techniques for Class I bulk sensor data.

Results of these and other analyses can in turn be directed towards developing architectural considerations in fulfillment of the objectives noted above.

12.1.2 Experimentation

12.1.2.1 Objectives and Emphasis

An experimental laboratory setup of a limited multi-node network should be established so that techniques may be tried and evaluated in software and hardware, and analytic tradeoffs verified quantitatively. Following are areas that would appear to benefit from experimental investigation in hardware and software. These are divided into several broad areas for clarity of presentation, but there is a direct relationship among all of the topics listed.

a. Software and Processor Architecture

1. Dynamic allocation of multiple voice rates
2. Movable boundary concept for both classes of traffic, and the allowable regions allocated as a function of processing and storage requirements
3. Linked list processing
4. Multiple addressing of data calls
5. Use of Class I procedures for Class II bulk sensor data.

b. Formats and Protocols

1. Protocol processing for various data interfaces
2. Common channel interoffice signaling (CCIS) processing, including dynamic and spill-backward alternate routing techniques
3. Packet processing, including ADCCP, for Class II data traffic and CCIS.

c. Interfaces

1. Functional and electrical characteristics of analog and digital secure and non-secure voice and data interfaces, towards assessing experimentally their impact on SENET-DAX architecture
2. Impact of satellite transmission delays
3. Data terminal throttling.

d. Synchronization

1. Master frame and network synchronization technique experimentation in hardware and software
2. Synchronization over satellite links.

e. Network Features

1. Techniques for voice and data precedence and preemption
2. Rudimentary traffic data collection and reduction.

12.1.2.2 Structure of the Experimentation Phase

The schedule and structure for the SENET-DAX initial experimentation phase should follow an approach similar to that for like programs. The major steps would be as follows:

a. Exact definition of the objectives and scope of the experiment(s), including number of "nodes" in the model, and voice and data terminals to be used
b. Required analysis towards the content of the experiment
c. Precise specification of equipment and software functions, operations, and structure to be designed; this includes description of voice and data terminal interfaces
d. Procurement, design, and integration of equipment and software for the model system
e. Formulation of test plan (scope, objectives, criteria for evaluation of results, schedule, etc.)
f. Generation of detailed test procedures
g. System and technique testing and evaluation
h. Test report and recommendations for further investigation.

12.1.3 Technical Interchanges

As noted above, a close relationship with DCA during the experimental phase is required, so that realistic goals can be set for the techniques investigations. Similarly, DCA must be involved in system test planning, to ensure that the test phase will reflect to their satisfaction verification of the system that was defined.

Another desirable goal is technical interaction with contractors pursuing allied study efforts, to have cross-fertilization of ideas during the techniques study portion of this new switching concept.

12.2 FAR-TERM RECOMMENDATIONS

Areas that may require more extended study and that do not appear to have as immediate an impact on SENET-DAX concept development as the near-term technique study areas mentioned above are:

a. Subscriber features
b. Message storage and retrieval
c. Time-assigned data interpolation (TADI)
d. System and Technical control
e. Man-machine interfaces
f. Comprehensive traffic statistics
g. Recovery modes.

There will undoubtedly be others. Some of these, such as recovery modes and subscriber features (conferencing in particular) can be initially considered in more detail in the near-term phase, but extended development can be deferred until later.

It should be added that a desirable area of investigation over both the near and the far term will be that of switch and network simulation. Specification of simulation requirements, design and coding for a large general purpose processor, and production runs, would be the steps in this development. Simulation will be a useful tool to evaluate tradeoffs, to sift out inapplicable techniques in advance of laboratory trials, and to evaluate performance of SENET-DAX switches designed according to the techniques investigated.

SECTION 13

REFERENCES

SECTION 13

REFERENCES

1. Aitcheson, E. and Cook, R., "No. 1 ESS ADF: Maintenance Plan", Bell System Technical Journal, Dec. 1970; pp 2831 - 2856.

2. Allen, D.W., "Statistics of Atomic Frequency Standards," Proc. IEEE, Vol. 54, Feb. 1966, pp 221 - 230.

3. American National Standards Institute, American National Standard for Advanced Data Communication Control Procedures (ADCCP), Draft 3, 13 February 1975.

4. Auerbach Associates Technical Report 2223-TR-2, "Automatic Store and Forward Switching Systems in the Field Army", 22 June 1973.

5. Barber, R.E., "Short-Term Frequency Stability of Precision Oscillators and Frequency Generators", Bell System Technical Journal, Vol. 50, March 1971, pp 881 - 915.

6. Barker, R.H., "Group Synchronizing of Binary Digital Systems", Communication Theory, Academic Press, London, 1953.

7. Barnes, J.A., "Atomic Timekeeping and the Statistics of Precision Signal Generators", Proc. IEEE, Vol. 54, pp 207 - 220.

8. Barnes, J.A., et. al., "Characterization of Frequency Stability", IEEE Trans. on I&M, Vol. 20, May 1971, pp 105 - 120.

9. Beneš, Mathematical Theory of Connecting Networks and Telephone Traffic, Chap. 4, "Strictly Non-Blocking Networks", 1965.

10. Benoit, D. and Cotton, I.W., "Prospects for Standardization of Packet Switched Networks", ANSI Task Group X3S37.

11. Binder, R., McGuillan, J. and Rettberg, R., "The Impact of Multi-Access Satellites on Packet Switching Networks", EASCON, 1975.

12. Blair, B.E., Ed., Time and Frequency: Theory and Fundamentals, National Bureau of Standards Monograph 140.

13. Boehn, W.B. and Mobley, R.L., "Adaptive Routing Techniques for a Distributed Communication System", IEEE Transactions on Comm. Tech., June, 1969.

14. Brandt, G.J. and Chretien, G.J., "Methods to Control and Operate a Message-Switching Network", Symposium on Computer-Communication Networks and Teletraffic, Polytechnic Institute of Brooklyn, April, 1972.

15. Browne, T., Wadsworth, D. and York, R., New Time Division Switch Units for No. 101 ESS, Bell System Technical Journal, Feb. 1969, pp 443 - 476.

16. Burton, H.O. and Sullivan, D.D., "Errors and Error Control", Proc. IEEE, Vol. 60, No. 11, Nov. 1972, pp 1293 - 1301.

17. Cohen, I., "Control of Data Processor Networks", Proc. IEEE ICC Conference, June, 1970.

18. Chu, W., "Optimal Message Block Size for Computer Communications with Error Detection and Retransmission Strategics", IEEE Transactions on Communications, Vol. COM-22, No. 10, October 1974, pp 1516 - 1525.

19. Coviello, G. and Vena, P., Concept for an Integrated Circuit - and Packet-Switched Telecommunications System, DCA Technical Comment No. 8-75, January 1975.

20. Coviello, G. and Vena, P., "Integration of Circuit/Packet Switching by a SENET (Slotted Envelope Network) Concept", NTC, New Orleans, 1975.

21. Defence Communications Agency, Advanced Planning Briefing for Industry, 25-26 April 1973, Title (U), Proceedings (S).

22. Defense Communications Agency, DCS AUTODIN Interface and Control Criteria, DCAC-370-D175-1.

23. Defense Communications Agency, AUTOVON System Interface Criteria, DCAC-370-V175-7.

24. DeWitt, R.G., "Network Synchronization Plan for the Western Union All-Digital Network", Telecommunications, 7 July 1973, pp 25 - 28.

25. Fischer and Harris, Analysis of an Integrated Circuit and Packet Switched Telecommunication System, DCEC Technical Note 6-75, January 1975.

26. Fitch, S. and Rechtenbaugh, D., Digital Data System: Testing and Maintenance, Bell System Technical Journal, May-June 1975, pp 845 - 860.

27. Frank, H., Kahn, R. and Kleinrock, L., "Computer Communication Network Design - Experience With Theory and Practice", Spring Joint Computer Conference, 1972.

28. Gray, J., "Line Control Procedures", Proc. IEEE, November 1972.

29. Hausenblas, A.E., "Notation of Communications Switching Operators in the Preprogramming Stage", International Switching Symposium, 1972.

30. Hellwig, H., "Atomic Frequency Standards: A Survey", 28th Annual Frequency Control Symposium, 1974, pp 315 - 339.

31. Hirota, K., Kato, M. and Yoshida, Y., A Design of Packet Switching System, NNT Corp., Tokyo, Japan.

32. Kartaschoff, P. and Barnes, J.A., "Standard Time and Frequency Generation", Proc. IEEE, Vol. 60, May 1972, pp 493 - 501.

33. Kleinrock, L. and Opderbeck, H., "Throughput in the ARPANET - Protocols and Measurement", Proceedings of the Fourth Data Communications Symposium, Quebec City, Canada, October 1975.

34. Kummerle, Multiplexer Performance for Integrated Line and Packet-Switched Traffic, IBM, Ruschlikon, Switzerland.

35. Mallory, P.E., "Methods of Frame Sync Analysis", Proceedings of the NTC, 1971, pp 264 - 269.

36. Martin, J., Design of Real Time Computer Systems, Prentice-Hall, Inc., Library of Congress No. 67-18923.

37. Martin, J., Telecommunications and the Computer, Prentice-Hall, 1969.

38. Martin, J., Teleprocessing Network Organization, Prentice-Hall, CH-5, 1970

39. Massey, J.L., "Optimum Frame Synchronization", IEEE Trans. on Comm., COM-20, April 1972, pp 115 - 119.

40. Maury, J.L. and Styles, F.J., "Development of Optimum Frame Synchronization Codes for Goddard Space Flight Center PCM Telemetry Standards", Proceedings of the NTC, 1964, pp 3-1-1 to 3-1-9.

41. McCoubrey, A.O., "A Survey of Atomic Frequency Standards", Proc. IEEE, Vol. 54, Feb. 1966, pp 116 - 135.

42. McGuillan, J. and Crowther, W., et. al., "Improvements in the Design and Performance of the ARPA Network", Fall Joint Computer Conference, 1972.

43. McKenzie, A., The ARPA Network Control Center, Fourth Data Communication Symposium, Quebec, Canada, Oct 1975, pp 5-1 to 5-6.

44. Meyer, R.F., Data Link Control Procedures Standards, National Telecommunications Conference Record, Vol. 1, November 1973.

45. Murphy, J.T., "Some Results on Frame Synchronization", IEEE Summer General Meeting, June 1963.

46. Naylor, W.E., "A Loop-Free Adaptive Routing Algorithm for Packet Switched Networks", Proceedings of the Fourth Data Communications Symposium, Quebec City, Canada, October 1975.

47. Neil, W., et. al., "Experimental Packet Switched Service: Procedures and Protocols", Part I, P.O. Elect. Eng. Journal, 1975.

48. Nielsen, P., "Some Optimum and Sub-optimum Synchronizers for Binary Data in Gaussian Noise", IEEE Trans. on Communications, June 1973, pp 770 - 772.

49. Pan, J.R., "Synchronizing and Multiplexing in a Digital Communications Network", Proc. IEEE, Vol. 60, No. 5, May 1972, pp 594 - 601.

50. Peavey, B., "Frame Synchronization in PCM Telemetry", Proceedings of the ITC, 1971, pp 316 - 323.

51. Pierce, J.R., "Synchronizing Digital Networks", Bell System Technical Journal, March 1969, pp 615 - 636.

52. Ricci, F.J., "The State of the Art and Control of Packet Switches and Networks", Proceedings of the ICC, August, 1975.

53. Rich, M.A., "Designing Phase-Locked Oscillators for Synchronization", IEEE Trans. on Comm., Vol. 22, July 1974, pp 890 - 896.

54. Roberts, L.G. and Wessler, B.D., "Computer Network Development to Achieve Resource Sharing", AFIPS Conference Proceedings 36, June 1970, pp 543 - 549.

55. Rocher, E.Y. and Pickholtz, R.L., "An Analysis of the Effectiveness of Hybrid Transmission Schemes", IBM Journal of Research and Development, July 1970, pp 426 - 433.

56. Saltzberg, B.R. and Zydney, H.M., "Network Synchronization", Bell System Technical Journal, Vol. 54, May-June 1975, pp 879 - 892.

57. Talley, D., Basic Telephone Switching Systems, Hayden Book Company, Inc., 1969.

58. Thomson, D., "Synchronization of an Integrated Digital Transmission and Switching Network, Post Office Electrical Engineer's Journal, 1975.

59. TRI-TAC, Data Adapter - Data Adapter and AN/TTC-39 - Data Adapter Interface Specification, TT-A3-4002-0015, 4 November 1974.

60. TRI-TAC, Performance Specification, Central Office, Communications, Automatic, AN/TTC-39 ( )(V), TT-B1-1101-0001A, 7 June 1974.

61. Wickler, M.R., "Path Delay, Its Variations, and Some Implications for the Field Use of Precise Frequency Standards", Proc. IEEE, Vol. 60, pp 522 - 529.

62. Wolman, E., "A Fixed Optimum Call Size for Records of Various Length", Journal of the ACM, 1965.

63 Zafiropolo, P., "Flexible Multiplexing for Networks Supporting Line Switched and Packet Switched Data Traffic", ICCC, Stockholm, Sweden, August 1974.

GTE Sylvania Internal Memoranda

64. Chernikoff, L., Block Parity for Trunk Signaling Messages, GTE Sylvania, 271-3000-622-254.

65. Toll, W.P., GPCO Common Channel Signaling Software Specifications, GPC-638-013, 2 July 1973.

66. TTC-39, Prototype Model Design Plan, Vol. 1, ECOM-0239-1, Oct. 1972.

67. TTC-39, Prototype Model Design Plan, Vol. 4, Section 2.

68. Chernikoff, L., Trunk Signaling Messages and Formats, Addendum 2 to Signaling Plan, TTC-39 PMDP, Vol. IV, 22 January 1973.

APPENDIX A

DAX TRAFFIC STATISTICS

APPENDIX A

DAX TRAFFIC STATISTICS

A.1 INTRODUCTION

An important factor in the calculation of many network variables (e.g., interswitch and cross-network delays, switch and trunk sizes, transmission overhead, Class I slot size, etc.) is the average subscriber generated traffic per node and link*. The aim of this appendix is to provide an estimate of this traffic and, just as importantly, to specify all major assumptions required to accomplish this aim.

The base statistics used to generate 1985 DAX subscriber originated traffic were provided by the DCA. The voice traffic base represents DCA's estimate of the 1976 switch-to-switch traffic for CONUS AUTOVON. The data traffic base is extracted from the "DoD Internet Study, Phase II Report" and represents their projection for 1985. Facsimile and video traffic are not included in either base but are estimated as described in Sections A.3.1 and A.3.2, respectively.

The results obtained in this appendix are highly dependent on the assumed network configuration and uniform traffic distribution. These assumptions were employed to ensure that symmetry could be used to simplify the analysis. Although the assumptions might not accurately reflect reality, they do provide valuable results because both assumptions yield overestimates of link and nodel traffic requirements and, therefore, provide an upper bound for these requirements.

*Network traffic due to CCIS messages, packet overhead, supervisory and control messages, etc. are not included. This aspect of network traffic is considered in Section 10.2.

A.2 VOICE TRAFFIC BASE

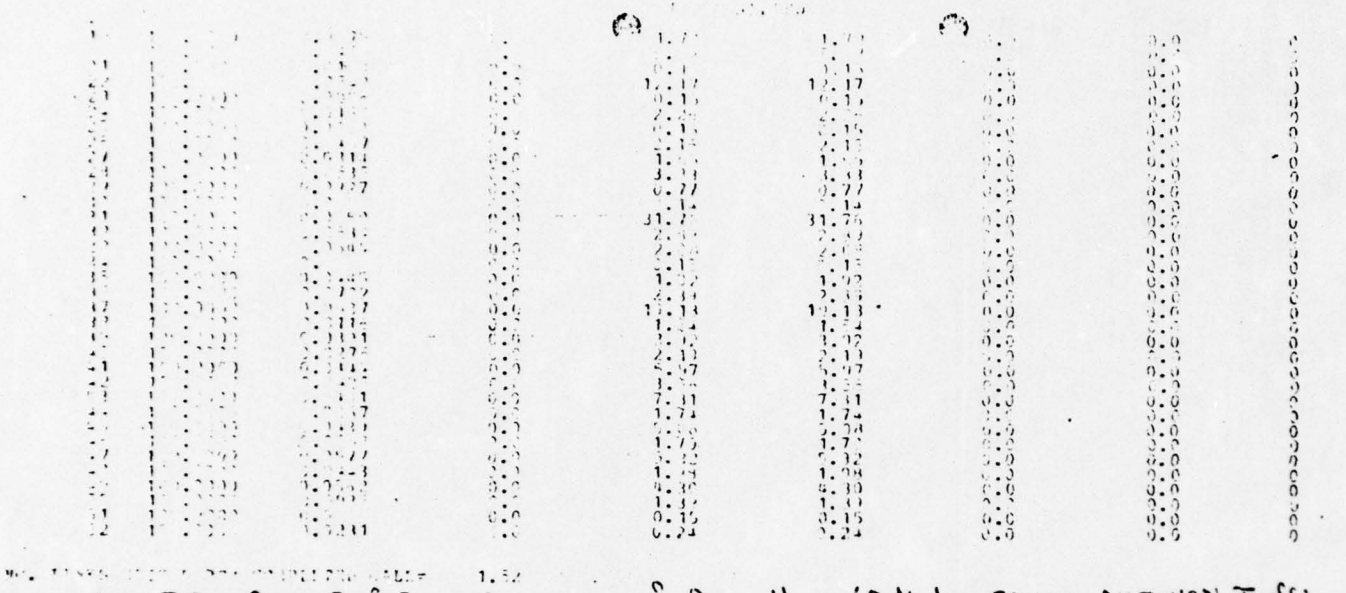
Table A-1 shows DCA's estimate of 1976 CONUS AUTOVON traffic broken down by node. The quantity of interest, total nodal traffic, is determined by summing for all nodes traffic to, traffic from, tandem traffic and pseudo traffic, and taking into account grade of service. The total nodal traffic is 10498E of which 2252E represent tandem traffic.

A.2.1 Data Traffic Base

Table A-2 shows the DOD's estimate of the 1985 data traffic distribution by terminal type. It will be necessary to modify this information in order to account for multiply addressed data messages. This will be accomplished by multiplying the final data traffic total by 1.75 which is the average number of addresses per message for the present AUTODIN.

TABLE A-1. VOICE TRAFFIC BASE

TABLE A-1. (CONT.)



Aug 75 Page 2 of 2 Summary of Busy Hour 60 Node CONUS AUTOVON Traffic

TABLE A-2. DATA TRAFFIC EXTRACTED FROM DOD DATA INTERNET STUDY PHASE II REPORT - DATA TRAFFIC BASE

	A Data Rate (Kb/s)	B Terminals	C one-way mess time (sec.)	(AxC) D message length (bits)	E mess/BH/ term	(DXE) F bits/BH term	(FxB) G bits/BH Total ($\times 10^9$)	(Gx7.5) H bits/Day Total ($\times 10^9$)	(BxE) I message/BH
1a. Low Speed Query	.15- .6(.$\overline{45}$)	17594	4	1.8K	20	36K	.633384	4.75038	351,880
1b. Low Speed Response	.15- .6(.$\overline{45}$)		40	18K	20	360K	6.33384	47.5038	351,880
2. High Speed Terminal	2.4 -4.8($\overline{3.6}$)	7836	15	54K	13	702K	5.50087	41.25654	101,868
3a. Computer Query	4.8	2560	.3	1.44K	130	.1872M	.479232	3.59424	332,800
3b. Computer Response	4.8		4	19.2K	175	3.36M	8.6016	64.512	448,000
3c. Bulk (1)	4.8		60	288K	15	4.32M	11.0592	82.944	38,400
3d. Bulk (2)	4.8		3600	17.28K	2 [1]		11.79648 [2]	88.4736	
							44.4046 x 10^9	333.03456 x 10^9	1,624,828

(1) per day
(2) 1/7.5 x G

A.3 1985 VOICE TRAFFIC

It is assumed that AUTOVON traffic will grow 2 percent per year between 1976 and 1985. Applying this factor to Table 1 gives a total 1985 BH nodal traffic of $10498 \times (1.02)^9 = 12546E$ of which $2252 \times (1.02)^9 = 2691E$ represent tandem traffic. The average BH voice traffic either terminating or originating at each note, subject to the following assumptions:

a. There are no pseudo nodes
b. Each node terminates and originates an equal amount of traffic.

is given by $(12546-2691)/(60x2) = 82.13E$

Two pertinent characteristics of AUTOVON can be derived from the above data. First, the total 1985 BH link traffic is given by

(total traffic + tandem traffic)/2 = (12546+2691)/2 = 7619E

And secondly, the average circuit length of an AUTOVON call is given by

total link traffic/total originated traffic =
7619/(12546-2691)/2 = 1.546 links/call

This last result is somewhat smaller than might be expected; however, it probably reflects the fact that CONUS AUTOVON is a distributed network with well-defined communities of interest.

In order to calculate the total call-bits represented by the above traffic, the following assumptions are made concerning voice calls:

a. An average hold time of 5 minutes
b. A call distribution by transmission rate given by - 10 percent 2400 b/s, 10 percent at 4000 b/s, 15 percent at 8000 b/s, 50 percent at 16,000 b/s, 10 percent at 32,000 b/s and 5 percent at 50,000 b/s - or weighted average voice transmission rate of 15,540 b/s
c. All calls are full-duplex.

Based on these assumptions, the BH call-bits transmitted or received by all nodes due only to call originations and terminations (neglecting tandem traffic) is calculated as follows:

a. 9855E (12546-2691) during the BH implies an average of 9855 call-originations each 5 minutes.

b. (9855 call-orig/5 min) x (12 5 min/BH) = 118,260 call-orig/BH

c. Call-bits/BH = call-orig/BH x avg. transmission rate x avg. hold time
= 118,260 x 15540 x 300
= 5.5132812×10^{11} bits/BH (or 4.1349609×10^{12} bits/day)

It should be stressed that this last result, 5.51328×10^{11}, is the total number of BH bits either transmitted or received by all nodes (excluding tandem traffic). The total nodal I/O is twice this value.

A.3.1 1985 Facsimile Traffic

In order to account for facsimile equipments, the high-speed terminal count listed in Table A-2 will be reduced by 10 percent and replaced by an equal number of facsimile terminals*. This reduces the number of high-speed terminals from 7836 to 7052 (-784 terminals) and reduces the number of bits/day produced by these terminals from 41.25654×10^9 to 37.13088×10^9 (-4.125654×10^9 bits).

As called for by the SOW, two types of facsimile service are required: one for terminals which operate in a continuous mode (Class I service) and the other for terminals with interruption capability (Class II service). Table A-3 summarizes the calculated traffic totals for the two classes of facsimile service. It should be noted that the effect of replacing 784 high-speed terminals by 784 facsimile terminals has been to increase the 1985 daily data traffic total by $(-4.125634 + 94.05 + 30.615) \times 10^9 = 120.539346 \times 10^9$ bits (from 333.03456×10^9 bits/day to 453.573906×10^9 bits/day).

*This approach was suggested by the DCA.

TABLE A-3. FACSIMILE TRAFFIC

	Class I Service	Class II Service
A. Average terminal rate	4.8 kb/s	50 kb/s
B. Average transmission time/transaction	7 min.	2/3 min
C. Transaction length (bits)	2×10^6	2×10^6
D. Number of terminals	627 (80%)	157 (20%)
E. Transactions/BH/ terminal	5 (1)	13 (2)
F. Error Control	FEC (1/2 rate)	ARQ
G. Bits/BH/terminal (CXE)	20×10^6 (3)	26×10^6
H. Bits/BH (G x D)	12.54×10^9	4.082×10^9
I. Bits/day (H x 7.5)	94.05×10^9	30.615×10^9
J. Transactions/BH (DXE)	3135	2041

(1) Assuming approximately 50 percent utilization of a facsimile terminal during the BH.

(2) Same as for high-speed terminals.

(3) Includes factor of 2 for FEC.

A.3.2 1985 Video Traffic

Because of the negligible impact video traffic is expected to have on the overall traffic load in 1985 and because of the lack of an historical base for this traffic, the following simplifying assumptions are made regarding video traffic:

a. An average hold time of 5 minutes
b. An average video terminal transmission rate of 256 kb/s
c. The average circuit length for a video call is the same as the average circuit length for a voice call, 1.546 links/call
d. A total nodal traffic (excluding tandem traffic) equal to 0.1 percent of the total nodal voice traffic, 9.855 E/BH.

Calculating the network call-bit impact of 9.855 E gives

$$9.855 \times 12 \times 256{,}000 \times 300 = 9.082368 \times 10^9 \text{ bits/BH (or } 68.11776 \times 10^9 \text{ bits/day)}$$

as the total bits either transmitted from or received at all nodes (excluding tandem traffic).

A.3.3 1985 Data Traffic

The 1985 subscriber traffic is as shown in Table A-2 with the exception that 784 high-speed terminals must be replaced by 784 facsimile terminals as described in Section A.3.1. The net effect on the totals given in Table A-2 is as shown below.

G. bits/BH total = 60.47652×10^9 (was 44.4046×10^9)
H. bits/day total = 453.573906×10^9 (was 333.03456×10^9)

A.4 DAX NETWORK STRUCTURE

The current thinking of the DCA is to have the DCS evolve into an integrated voice/data network possessing a modified hierarchical structure. To this end, the most suitable DAX network structure to assume for the purposes of this traffic analysis is the singly spoked wheel configuration shown in Figure A-1. This configuration is characterized by good survivability (each node is connected to at least three other nodes) and short cross-network delays (no primary path between any two nodes requires more than three tandem links). It should be noted that the DAX is only being considered for use as an access or regional node. However, to simplify the traffic analysis, it will be assumed that there will be DAX's at all nodes.

As shown in Figure A-1, each tandem node is connected to every other tandem node and to at most INT ((60-n)/n) + 1 regional nodes*; each regional node is connected to two other regional nodes and to one tandem node. Another configuration which might merit future consideration is the doubly spoked wheel, where each regional node is homed on two tandem nodes. This structure has very good survivability but requires (60-n) more links that the singly homed case. The doubly spoked wheel will not be considered here.

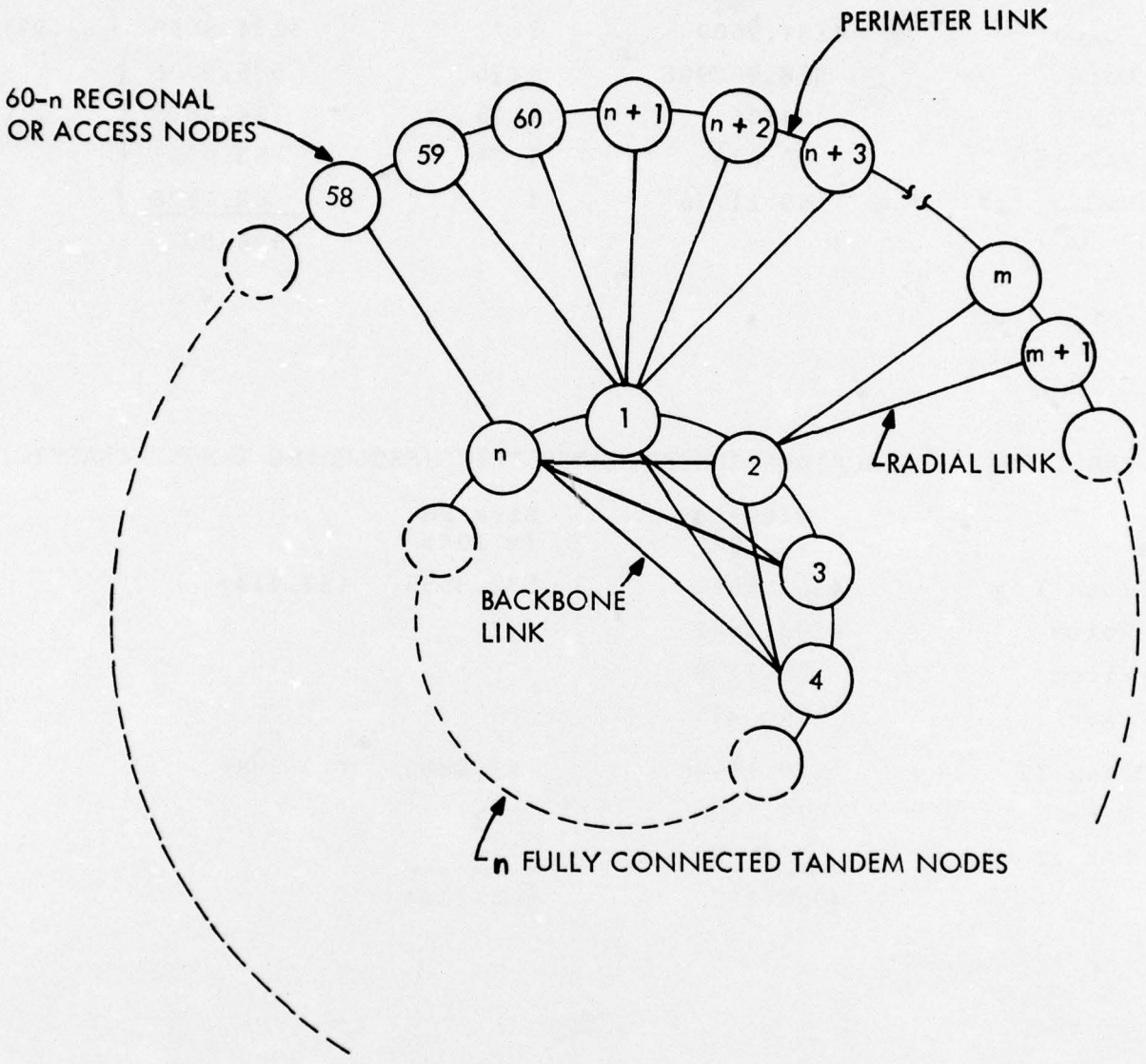
A.4.1 DAX Subscriber Traffic

The total number of bits transmitted (or received) daily from the 60 DAX nodes due to terminating and originating traffic only (excluding tandem traffic) is given in Table A-4. As may be seen the effect of multiple addressing is included. Table A-5 shows the distribution of this traffic by class of service.

A.4.2 DAX Network Traffic

Network traffic is made up of both subscriber and tandem traffic. The quantity of subscriber traffic impinging on the DAX network is as specified in Section A.4.1. Tandem traffic is a complicated function of

*It is assumed that future DAX network nodes will be restricted to the 60 CONUS AUTOVON locations.



REGIONAL NODES: 60-n
TANDEM NODES: n
TOTAL: 60

PERIMETER LINKS: 60-n
RADIAL LINKS: 60-n
BACKBONE LINKS: $n(n-1)/2$
TOTAL: $120 - 5n/2 + n^2/2$

9373-75E

Figure A-1. Singly Spoked Wheel Configuration

TABLE A-4. SUBSCRIBER-GENERATED TRAFFIC (EXCLUDING TANDEM TRAFFIC)

Category	Bits/Day ($\times 10^9$)	Multiple Message Factor	Bits/Day ($\times 10^9$)	
Voice	4134.9609	1	4134.9609	(82.8%)
Data	328.908906	1.75	575.5906	(17.2%)
FAX-I	94.05	1.75	164.5875	
FAX-II	30.615	1.75	53.5763	
Video	68.11776	1	68.1178	
			4996.8331	

TABLE A-5. SUBSCRIBER-GENERATED TRAFFIC (EXCLUDING TANDEM TRAFFIC)

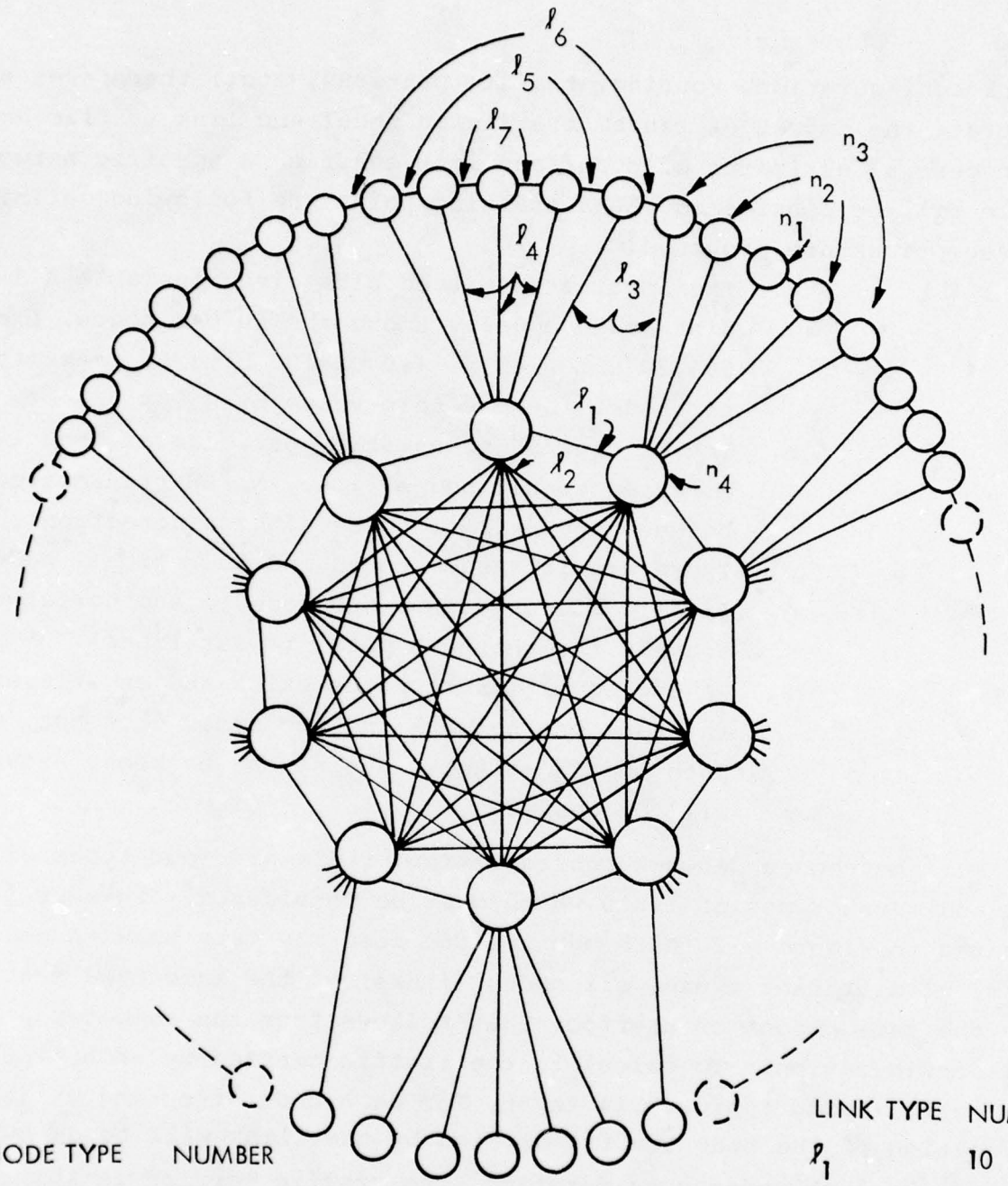
	Bits/Day ($\times 10^9$)	Bits/BH ($\times 10^9$)	
Class I	4367.6662	582.3555	(87.41%)
Voice	4134.9609		
Video	68.1178		
FAX-I	164.5875		
Class II	629.1669	83.8889	(12.59%)
Data	575.5906		
FAX-II	53.5763		
	4996.8331	666.2444	

network configuration, routing, traffic patterns, etc.; therefore, to illustrate the impact of tandem traffic on nodal and link traffic and to obtain general estimates of nodal and link traffic, a specific network example will be considered. As a starting point the following definitions and assumptions are proposed:

a. The total transmitted bits given in Table A-5 distribute equally among the 60 DAX nodes, i.e., $666.24 \times 10^9/60 = 11.04 \times 10^9$ bits/BH transmitted per node. Denote this value by N_T.
b. The N_T bits/BH transmitted per node are equally directed to all nodes, i.e., N_T/BH transmitted between any pair of nodes (in one direction).
c. There are 10 tandem nodes and 50 regional nodes.
d. Traffic directed from one node to another always travels the shortest path (fewest links).
e. If two equal length paths exist and one traverses the backbone network and the other does not, the path that does not traverse the backbone network will be chosen.

For the chosen network configuration, there are four types of nodes and seven types of links which must be considered. This is illustrated in Figure A-2. Although traffic load may vary between nodes (links) of different types, all nodes (links) of the same type must carry the same amount of traffic. This follows from the symmetry of the wheel configuration. To calculate the traffic carried by each type of link the following approach is taken. For each node affecting a link*, the fraction of the node traffic carried by that link will be calculated. Then for the link under consideration, the traffic carried is summed over all nodes by type. The different link traffic due to the four types of nodes is shown in Figure A-3a through A-3d. In each case, the darkened node is the node transitting data. Using these figures, the total link traffic by type is calculated as follows:

*A node affects a link if it causes traffic to be carried on that link.



NODE TYPE	NUMBER
n_1	10
n_2	20
n_3	20
n_4	10
	60

LINK TYPE	NUMBER
ℓ_1	10
ℓ_2	35
ℓ_3	20
ℓ_4	30
ℓ_5	20
ℓ_6	10
ℓ_7	20
	145

9375-75E

Figure A-2. Wheel Configuration - 10 Tandem Nodes/50 Access Nodes

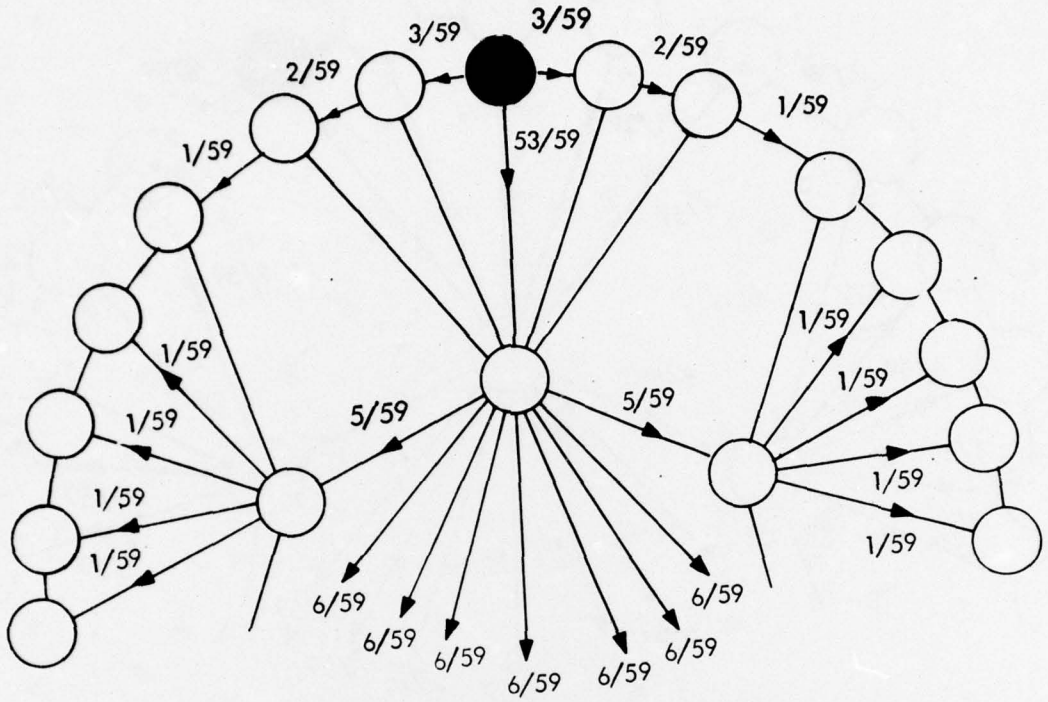


Figure A-3a. Type 1 Node Transmitting

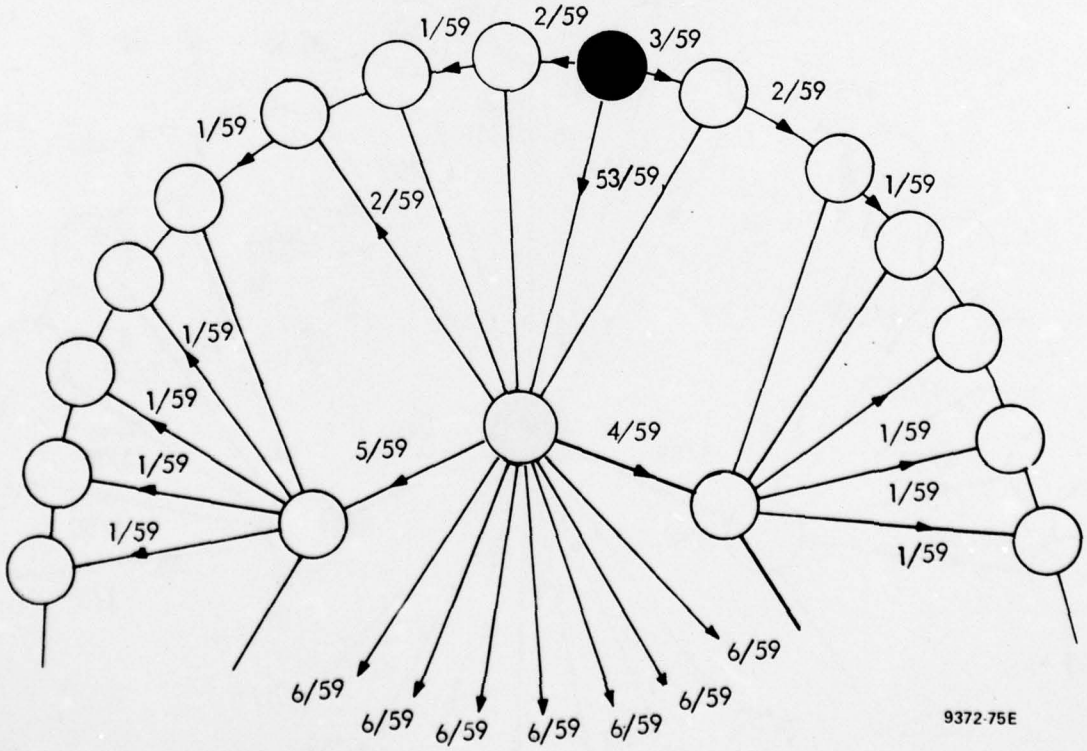


Figure A-3b. Type 2 Node Transmitting

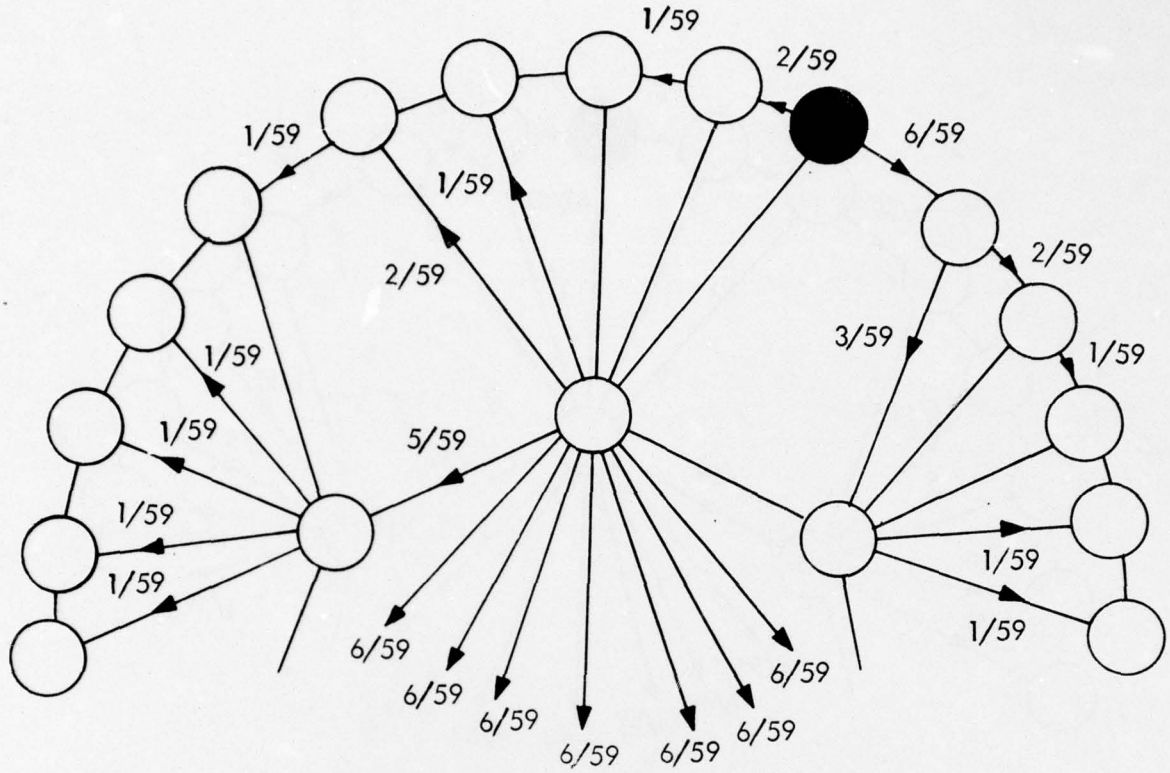


Figure A-3c. Type 3 Node Transmitting

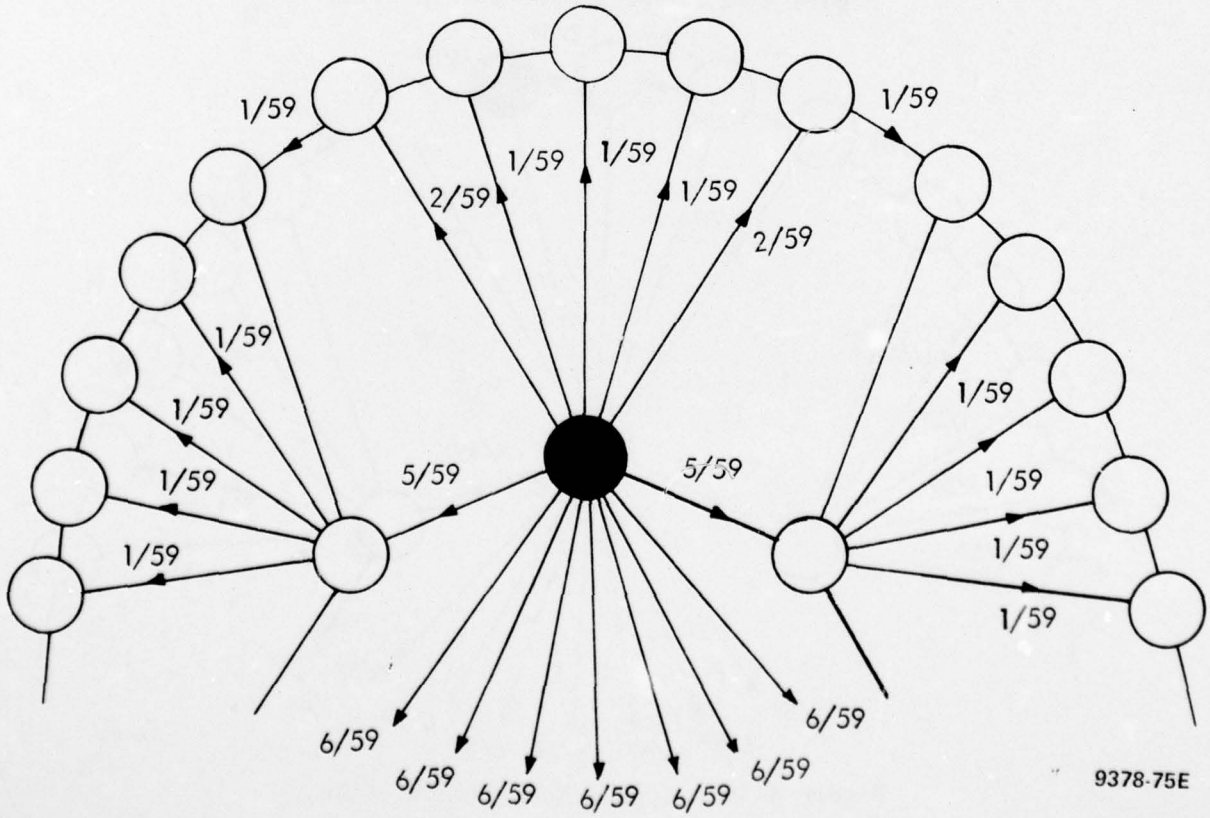


Figure A-3d. Type 4 Node Transmitting

l_1: $N_T(5/59 + 4/59 + 5/59 + 0 + 5/59 + 5/59) = 24/59\ N_T$

l_2: $N_T(6/59 + 6/59 + 6/59 + 6/59 + 6/59 + 6/59) = 36/59\ N_T$

l_3: $N_T(51/59 + 3/59) = 54/59\ N_T$

l_4: $53/59\ N_T$

l_5: $N_T(2/59 + 3/59) = 5/59\ N_T$

l_6: $N_T(1/59 + 2/59 + 1/59 + 1/59 + 6/59 + 1/59) = 12/59\ N_T$

l_7: $N_T(3/59 + 1/59) = 4/59\ N_T$

From these results, the total nodal traffic by type is given as follows:

n_1: $2l_7 + l_4 = (2(4/59) + 53/59)\ N_T = 61/59\ N_T$

n_2: $l_4 + l_5 + l_7 = (53/59 + 5/59 + 4/59)\ N_T = 62/59\ N_T$

n_3: $l_3 + l_5 + l_6 = (54/59 + 5/59 + 12/59)\ N_T = 71/59\ N_T$

n_4: $2l_1 + 7l_2 + 2l_3 + 3l_4 = ((2(24/59) + 7(36/59) + 2(54/59) + 3(53/59))\ N_T = 567/59\ N_T$

From the above results, the following traffic limits are obtained:

	Maximum ($\times 10^9$ bits/BH)	Minimum (10^9 bits/BH)
Radial Link:	10.1	9.92
Perimeter Link:	2.25	.75
Backbone Link:	6.74	4.49
Regional Node:	13.29	11.41
Tandem Node:	106.1	106.1

It should be noted that the last result includes tandem traffic. Several interesting facts are now apparent, they are as follows:

(1) The radial links carry more traffic than the backbone links. However, as the number of tandem nodes decreases, it is expected that the disparity between the two would also decrease.

(2) The capacity of a T1 line in bits/BH is $1.544 \times 10^6 \times 3600 = 5.56 \times 10^9$. For the given network design, a T1 line is insufficient for use as a radial link and possible for use as a backbone link.

(3) A doubly spoked wheel configuration would greatly reduce the radial link load, probably, by a factor of 2.

(4) The total nodal bits transmitted during the BH (including tandem traffic) is given by

$$10n_1 + 20n_2 + 20n_3 + 10n_4 = 151.52\ N_T = 1.6728 \times 10^{12} \text{ bits/BH}$$

(5) The average length of a circuit is

$$151.52\ N_T/60\ N_T = 2.53 \text{ links}$$